Universidade de Aveiro
Departamento de
Eletrónica, Telecomunicações e Informática
2017

João Pedro Brites Ferreira Nogueira

Entrega de Conteúdos Multimédia em Over-The-Top: Caso de Estudo das Gravações Automáticas

Over-The-Top Multimedia Delivery: A Catch-Up TV Case Study

Universidade de Aveiro
Departamento de
Eletrónica, Telecomunicações e Informática
2017

João Pedro Brites Ferreira Nogueira

Entrega de Conteúdos Multimédia em Over-The-Top: Caso de Estudo das Gravações Automáticas

Over-The-Top Multimedia Delivery: A Catch-Up TV Case Study

Tese apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Engenharia Eletrotécnica, realizada sob a orientação científica da Doutora Susana Isabel Barreto de Miranda Sargento, Professora Associada com Agregação do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e coorientação do Doutor Lucas Guardalben, Investigador de Pós-Doutoramento do Instituto de Telecomunicações de Aveiro.

O trabalho desenvolvido no decorrer desta Tese foi financiado pela Bolsa de Doutoramento SFRH/BD/51565/2011, pela Fundação para a Ciência e Tecnologia (FCT), pelo projeto IAPMEI, QREN e COMPETE GAPOTT (Gravações Automáticas e Publicidade Over-The-Top) N.º 2013/34009, projeto Portugal 2020 UltraTV (POCI-01-0247-FEDER-017738), e pela Altice Labs, SA.

o júri
the jury

presidente
president

Carlos Manuel Martins da Costa
Professor Catedrático da Universidade de Aveiro

vogais
examiners committee

Alexandre Júlio Teixeira Santos
Professor Associado com Agregação, Escola de Engenharia, Universidade do Minho

João Paulo Silva Machado Garcia Vilela
Professor Auxiliar da Faculdade de Ciências e Tecnologia, Universidade de Coimbra

Ricardo Santos Morla
Professor Auxiliar da Faculdade de Engenharia da Universidade do Porto

Diogo Nuno Pereira Gomes
Professor Auxiliar da Universidade de Aveiro

Susana Isabel Barreto Miranda Sargento
Professora Associada com Agregação da Universidade de Aveiro (orientadora)

agradecimentos
acknowledgements

O presente trabalho não teria sido possível sem a colaboração e apoio de vários familiares, amigos, colegas, orientadora e co-orientador, pelo que agradeço a todos os que, direta ou indiretamente, me auxiliaram nos últimos 5 anos.

Deixo também um agradecimento especial aos seguintes coordenadores que, não sendo meus orientadores formais, me apoiaram nesta jornada.

Mestre José Bernardo dos Santos Cardoso
Departamento de Digital, Internet e Televisão, Altice Labs SA, Portugal

Professor Doutor Jorge Ferraz de Abreu
Departamento de Comunicação e Arte, Universidade de Aveiro, Portugal

Professor Doutor Valdecir Becker
Departamento de Sistemas de Computação, Universidade Federal da Paraíba, Brasil

Professor Doutor Peter Steenkiste
Computer Science Department, Carnegie Mellon University, Pittsburgh, USA

Palavras-chave

Over-The-Top, Multimédia, Gravações Automáticas, Redes de Entrega de Conteúdos, *Caching*, Qualidade de Experiência, *Machine Learning*, Conhecimento dos Conteúdos

Resumo

A entrega de conteúdos multimédia em Over-The-Top (OTT) é uma proposta atrativa para fornecer um serviço flexível e globalmente acessível, capaz de alcançar qualquer dispositivo, com uma promessa de baixos custos. Apesar das suas vantagens, é necessário um planeamento arquitetural detalhado e otimizado para manter níveis elevados de Qualidade de Experiência (QoE), em particular aquando da migração dos serviços suportados em redes geridas com garantias de qualidade pré-estabelecidas. Para colmatar a falta de trabalhos de investigação na área de sistemas de entrega de conteúdos multimédia em OTT, esta Tese foca-se na otimização destas soluções como um todo, partindo do caso de uso de migração de um serviço popular de Gravações Automáticas suportado em redes de Televisão sobre IP (IPTV) geridas, para um cenário de entrega em OTT. Um estudo global para aferir a importância das Gravações Automáticas revela a sua relevância no panorama de serviços multimédia e a sua adequação enquanto caso de uso de migração para cenários OTT. São obtidos registos de consumos de um serviço de produção de Gravações Automáticas, representando mais de 1 milhão de assinantes, para caracterizar e extrair informação de consumos numa escala e âmbito não contemplados até à data na literatura. Esta caracterização é utilizada para contruir modelos de previsão de carga, tirando partido de sistemas de *machine learning*, que permitem otimizações estáticas e dinâmicas dos sistemas de entrega de conteúdos em OTT através de previsões das necessidades de largura de banda e armazenamento, potenciando ganhos significativos em consumo energético e custos. Um novo mecanismo de *caching*, *Most Popularly Used (MPU)*, demonstra um desempenho superior às soluções de referência, quer em cenários de simulação quer experimentais. A necessidade de medição exata da QoE em *streaming* adaptativo HTTP motiva a criação de um modelo capaz de endereçar aspetos específicos destas tecnologias adaptativas. Ao endereçar a cadeia completa de entrega através de uma arquitetura consciente dos seus conteúdos, esta Tese demonstra que são possíveis melhorias de desempenho muito significativas nas redes de entregas de conteúdos em OTT de próxima geração.

Keywords Over-The-Top, Multimedia, Catch-up TV, Content Delivery Networks, Caching, Quality of Experience, Machine Learning, Content-Awareness

Abstract Over-The-Top (OTT) multimedia delivery is a very appealing approach for providing ubiquitous, flexible, and globally accessible services capable of low-cost and unrestrained device targeting. In spite of its appeal, the underlying delivery architecture must be carefully planned and optimized to maintain a high Quality-of-Experience (QoE) and rational resource usage, especially when migrating from services running on managed networks with established quality guarantees. To address the lack of holistic research works on OTT multimedia delivery systems, this Thesis focuses on an end-to-end optimization challenge, considering a migration use-case of a popular Catch-up TV service from managed IP Television (IPTV) networks to OTT. A global study is conducted on the importance of Catch-up TV and its impact in today's society, demonstrating the growing popularity of this time-shift service, its relevance in the multimedia landscape, and fitness as an OTT migration use-case. Catch-up TV consumption logs are obtained from a Pay-TV operator's live production IPTV service containing over 1 million subscribers to characterize demand and extract insights from service utilization at a scale and scope not yet addressed in the literature. This characterization is used to build demand forecasting models relying on machine learning techniques to enable static and dynamic optimization of OTT multimedia delivery solutions, which are able to produce accurate bandwidth and storage requirements' forecasts, and may be used to achieve considerable power and cost savings whilst maintaining a high QoE. A novel caching algorithm, Most Popularly Used (MPU), is proposed, implemented, and shown to outperform established caching algorithms in both simulation and experimental scenarios. The need for accurate QoE measurements in OTT scenarios supporting HTTP Adaptive Streaming (HAS) motivates the creation of a new QoE model capable of taking into account the impact of key HAS aspects. By addressing the complete content delivery pipeline in the envisioned content-aware OTT Content Delivery Network (CDN), this Thesis demonstrates that significant improvements are possible in next-generation multimedia delivery solutions.

Contents

List of Figures

List of Tables

Listings

Acronyms

3GPP 3rd Generation Partnership Project

AAA Authentication, Authorization and Accounting

AAC Advanced Audio Coding

AES Advanced Encryption Standard

AIR Adobe Integrated Runtime

AP Access Point

ARPU Average Revenue Per User

ARR Application Request Routing

ATS Apache Traffic Server

AVC Advanced Video Coding

BA Business Analytics

BI Business Intelligence

BRNN Bayesian Regularized Neural Network

BSS Business Support System

CAPEX Capital Expenditures

CARP Cache Array Routing Protocol

CDN Content Delivery Network

CDF Cumulative Distribution Function

CI Confidence Interval

CPU Central Processing Unit

CRISP-DM Cross Industry Standard Process for Data Mining

CRM Customer Relationship Management

DASH-IF DASH Industry Forum

DASH Dynamic Adaptive Streaming over HTTP

DECE Digital Entertainment Content Ecosystem

DHT Distributed Hash Table

DNS Domain Name System

DoS Denial of Service

DPI Deep Packet Inspection

DRM Digital Rights Management

DSL Digital Subscriber Line

DVR Digital Video Recorder

EPG Electronic Programming Guide

ESI Edge Side Includes

EST-VoD Electronic Sell Through VoD

EU European Union

FCAN Flash Crowds Alleviation Network

FIFO First-In-First-Out

Fps Frames per second

FTTH Fiber-To-The-Home

GMT Greenwich Mean Time

GOP Group of Pictures

GSLB Global Server Load Balancing

HAS HTTP Adaptive Streaming

HDD Hard Disk Drive

HDS HTTP Dynamic Streaming

HD High-Definition

HLS HTTP Live Streaming

HTCP Hypertext Caching Protocol

HTTP Hypertext Transfer Protocol

ICN Information-Centric Networks

ICP Internet Cache Protocol

IDS Intrusion Detection System

IETF Internet Engineering Task Force

IIR Infinite Impulse Response

IIS Internet Information Services

ILP Integer Linear Programming

IoT Internet-of-Things

IPTV IP Television

IP Internet Protocol

ISP Internet Service Provider

IT Information Technologies

ITU-T International Telegraph Union Telecommunication Standardization Sector

ITU-R International Telegraph Union Radiocommunication Sector

IV Initialization Vector

k-HST k-Hierarchically well-Separated Trees

k-NN k-Nearest Neighbor

KQI Key Quality Indicator

LFN Long Fat Network

LFU Least Frequently Used

LFU-W LFU-Weighted

LIRS Low Inter-reference Regency Set

LRU Least Recently Used

LRU-W LRU-Weighted

LOOCV Leave-One-Out Cross-Validation

LTE Long Term Evolution

MAE Mean Absolute Error

MASE Mean Absolute Scaled Error

MB MegaBytes

MCNKP Multiple-Choice Nested Knapsack Problem

MDC Multiple Description Coding

MILP Mixed Integer Linear Programming

MON Managed Operator Network

MOS Mean Opinion Score

MPD Media Presentation Description

MPEG-2 TS MPEG-2 Transport Stream

MPEG Moving Pictures Expert Group

MPU Most Popularly Used

MSE Mean Squared Error

NNet Neural Network

NPVR Network Personal Video Recorder

NP Non-deterministic Polynomial-time

NZV Near-Zero Variance

OPEX Operational Expenditures

OS Operating System

OSS Operations Support System

OTA Over-The-Air

OTT Over-The-Top

P2P Peer-to-Peer

PC Principal Component

PCA Principal Component Analysis

PEVq Perceptual Evaluation of Video Quality

PID Proportional-Integral-Derivative

PIFF Protected Interoperable File Format

PLS Partial Least Squares

PoP Point-of-Presence

PPV Pay-Per-View

PSNR Peak Signal-to-Noise Ratio

PSQA Pseudo-Subjective Quality Assessment

PVR Personal Video Recorder

QMON Quality Monitoring

QoE Quality-of-Experience

QoS Quality-of-Service

RAM Random Access Memory

RAN Radio Access Network

RD Rate–distortion

RF Random Forest

RMSE Root Mean Squared Error

RNN Random Neural Network

RTCP RTP Control Protocol

RTP Real-time Transport Protocol

RTSP RTP Streaming Protocol

RTT Round Trip Time

SDK Software Development Kit

SD Standard Definition

SLA Service Level Agreement

SNMP Simple Network Management Protocol

SSIM Structural Similarity

SSL Secure Sockets Layer

STB Set-Top-Box

STSQ Short-Time Subjective Quality

SVC Scalable Video Coding

SVM Support Vector Machine

S-VoD Subscription VoD

TASE Training Average Scaled Error

TCP Transmission Control Protocol

TVSQ Time-Varying Subjective Quality

T-VoD Transaction VoD

UDP User Datagram Protocol

URL Uniform Resource Locator

USA United States of America

VM Virtual Machine

VoD Video-on-Demand

WMA Windows Media Audio

XML Extensible Markup Language

Chapter 1

Introduction

This chapter presents a holistic overview of the topics encompassed in modern OTT multimedia networks and details the full range of issues that inspire this Thesis. After the initial motivation section, the research goals and approach are laid down with the purpose of clearly identifying the scientific domains under study. The main contributions are subsequently summarized and followed by a global Thesis outline.

1.1 Motivation

The key technological innovation and growth driver of the past decade has been, beyond any doubt, the Internet, acting as a catalyst to the ongoing third industrial revolution, or the *Information Age* [12]. The ever increasing device connectivity has motivated large scale investments on research, communication infrastructures, network technologies, and Information Technologies (IT) development up to unprecedented levels that were never expected, even in the most optimistic earlier predictions.

The Internet has evolved from an academic domain into business and home environments, and is now so popular that it is on the way to be considered a commodity, or even a fundamental human right as a means to free information access.

While the initial development drivers were commercial in nature and strongly tied to telecommunication operators, the widespread adoption of the Internet exposed it to an increasing number of users that generated benefits to the Internet community as a whole, resulting on an ever increasing number of services that take advantage of the direct, and often personal, communication channels provided by the Internet to offer value-added paid services, as is the case of music and video streaming services, but also of other traditional services with physical media distribution, such as newspapers or books.

As a complement to the fast-paced evolution on server and network technologies, client terminals have also witnessed revolutionary developments in the past decade, up to a point where the number of connected devices outnumber the amount of inhabitants on most developed countries with affordable Internet access, which ultimately led to the concept of Internet-of-Things (IoT), where people live surrounded by smart Internet-enabled devices responsible for continuous information retrieval and processing.

The rise of Over-The-Top (OTT) services.

The number of OTT services, characterized by being transmitted over any network without operators' control in the distribution process, has been on the rise. Supported by an existing and usually free Internet backbone, service providers have exploited the characteristics of unmanaged networks to deliver services to their consumers without having to invest heavily in infrastructure. Due to their nature, OTT services have an inherent widespread reachability, and are able to accommodate virtually any Internet-connected device, without requiring network-specific equipment or management capabilities, as opposed to traditional managed services requiring specific network support.

The recent approval of network-neutrality laws in the United States of America (USA) and European Union (EU) [13, 14], restricts the Internet Service Providers (ISPs)' ability to block or throttle Internet traffic with the purpose of preventing discrimination and increasing consumer choice. Exceptions must be due to compliance with legal obligations, ensuring network integrity, and congestion management in exceptional and temporary situations. The mandated equal treatment for Internet traffic ensures that new OTT services have a fair chance of competing with operator-provided services.

The network-neutrality legislation is particularly relevant for multimedia content, which is responsible for an outstanding amount of OTT traffic, as can be shown by the tremendous popularity of YouTube, Skype, and Netflix to name a few. Consumers crave live TV, Catch-up TV, and on-demand video with high quality, available in real time, and without geographic or technological restrictions. Cisco [15] estimates that approximately 64% of Web traffic in 2014 was due to video content, and the global revenue from OTT multimedia services is expected to reach 19 billion USD by 2018 [16], up from 9 billion USD in 2014. In spite of the advantages of providing services in an OTT model, several challenges exist. Scalability, devices' heterogeneity, and Quality-of-Experience (QoE) are key concerns for OTT service providers.

Scalability issues arise due to two main reasons: internal limitations and external limitations. Regarding the internal limitations, the service must be designed with scale-out capabilities in order to be able to grow in capacity so that growths in demand may be accommodated by adding additional computing, memory, storage, or network resources. As for external limitations, the uncontrolled nature of the delivery infrastructure may lead to situations where the uncontrolled network infrastructure does not have the adequate resources to handle the necessary network traffic. This limitation was evident in a recent tussle between Netflix and Verizon [17], where Verizon required Netflix to pay a fee for their egress traffic in order to avoid congestion.

As for heterogeneity concerns, the challenge is to provide services able to support a wide range of operating systems (iOS, Android, Windows, Linux, ...), network access technologies (Fiber-To-The-Home (FTTH), Wi-Fi, Long Term Evolution (LTE), ...) and devices with different capabilities such as display resolution, and processing power.

Lastly, the focus on QoE has risen in the past few years, driven by increased users' expectations and desire for high quality content with immediate availability. In order to compensate for a lack of proper modeling of QoE on OTT services, some service providers rely on heuristics that add subjective components to technical QoS parameters;

however, these approaches are far from ideal and fall short of providing a great user experience. For example, Pay-TV providers custom tailor their content streams to the clients' platforms, often serving lower bit-rate content to mobile devices, regardless of their actual processing capabilities. Clearly, considering the heterogeneity issues previous mentioned, this coarse-grained approach fails to maximize the QoE for consumers with good mobile terminals. These issues have, in part, been solved by adaptive streaming technologies, which in spite of being a step on the right direction still do not objectively address the issue of QoE maximization, which is now seen as a major differentiator between OTT services, and can make or break the success of an OTT-based service provider [18], particularly in the case of Pay-TV services.

Notwithstanding these concerns, in part related to the incipient nature of these technologies, research efforts have been conducted in several fronts, ranging from terminal improvements, up to network awareness and additional service intelligence with the goal of adapting the offered services to current resources' limitations concerning network and device capabilities. These challenges have been identified as targets to address by European research projects [19, 20, 21]. One of the major outcomes of these efforts towards video delivery optimization OTT was the standardization of DASH [22].

OTT is reshaping the video industry.

The importance of OTT services in the industry may be observed by a shift in paradigm of the viewers' usage patterns: with respect to TV broadcasting services, there is a clear increasing trend of non-linear TV video watching; regarding Video-on-Demand (VoD) consumption, most of the traditional movie rental stores have closed and either focused on online video delivery (e.g. Netflix), or perished (e.g. Blockbuster).

The multimedia OTT market is a multi-million dollar industry, where market players fight for dominance of users' viewing time. A competitive OTT service must address customers' requirements and the supply chain requirements. Customers' requirements encompass multi-screen support, rich user interfaces, flexible pricing, integration with social media, broad content catalog, access anytime/anywhere service, and high QoE. On the other hand, supply chain requirements focus on content rights, access to distribution channels, infrastructure and devices. The ability to address these requirements determines the success of the service as a whole.

Given the multitude of parties involved in the OTT service delivery process, several competing powers rise naturally. Figure 1.1 exhibits the most prominent parties involved in OTT services. On the one hand, content providers - such as TV broadcasters and studios - perceive OTT streaming as an opportunity to bypass telecommunication companies and serve the content directly to consumers; thus, controlling the entire service chain. Examples of this approach include sports TV channels and movies studios.

On the other hand, telecommunications companies act as service aggregators with convergent offers including dual, triple, four, and fifth-play services, and control the delivery infrastructures required to reach consumers. Moreover, they often have the additional benefit of owning a CDN, improving users' QoE due to their closely located servers.

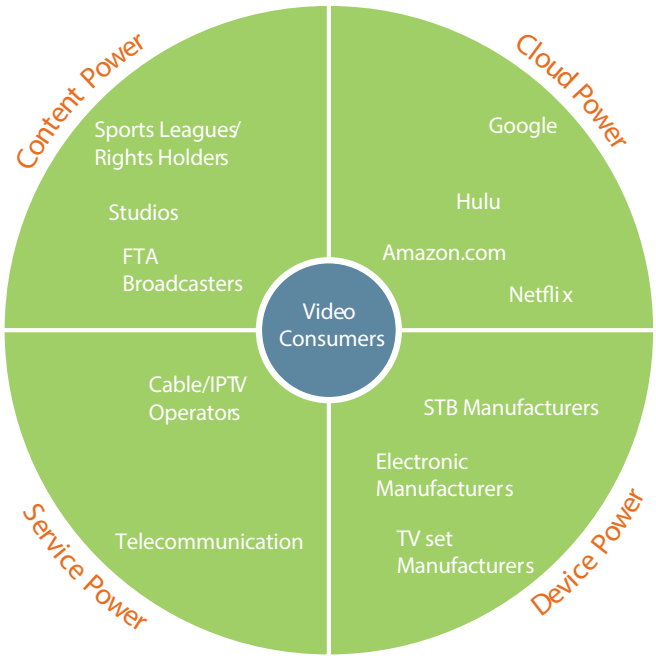


Figure 1.1: OTT competition framework [1].

Device manufacturers represent a third party with the ability to pre-load devices with the software of their choice and must be taken into account.

Finally, cloud players, such as Netflix and Amazon Instant Video, are perfect examples on how cloud-based service providers also take advantage of the OTT delivery model to sell their own offers.

Perspective for Telecommunications Operators.

Telecommunications operators, particularly those which also provide IPTV offers, often perceive OTT providers as threats; however, they are in a unique position to expand their businesses by migrating valued-added services to OTT, leveraging this new delivery model to improve their value proposition and, ultimately, increase their revenue.

A head-first approach to OTT business models may drive operators into developing their own OTT services as competitors to other big and already established players, which can prove problematic: how easy would it be to compete with Netflix? A smarter alternative would be to offer competing services where good chances for success exist – e.g. live TV and Catch-up TV –, while partnering with dominant services and content providers where competing is not viable and/or profitable – e.g. Skype or YouTube. Instead of following a dumb-pipe model and simply routing content to end-users, a telecommunications operator may leverage its client proximity and information to add value to an OTT business relationship.

Telecom companies hold several valuable assets and competences of interest to ex-

ternal service and content providers, such as [23]: a large client base with advanced Customer Relationship Management (CRM) solutions in place; valuable client information including service profiles and identities; real-time awareness of user-context – e.g. location and device; network monitoring, management and control.

Their proximity to end-users is favorable when compared with global OTT providers, and enables optimizations not available to outside competitors, such as placing caches at the edge aggregation points, improving content locality, and fine-tuning the flow of data in order to minimize congestion and maximize users' QoE.

These characteristics turn an apparently dispensable part of the delivery pipeline into a valuable partner, capable of capturing the interest of service and content providers.

OTT delivery is also appealing as an evolution to existing services currently delivered through managed networks, as is the case with IPTV services. By migrating these services into an OTT delivery model, convergence is easier to achieve [24]. As an example, an OTT multi-screen IPTV service can easily target clients inside or outside managed networks, which is especially useful for advanced content delivery features such as Network Personal Video Recorder (NPVR), Catch-up TV, and VoD.

Summary

OTT services have been growing at a fast pace driven by a low barrier of entry, mostly because of little to no investment being required in infrastructures traditionally necessary to reach the masses. This fast-paced growth presents an opportunity for all the involved partners, but comes with several challenges, especially with regard to scalability and QoE, which must be addressed.

In the face of being treated as dumb-pipes, current telecommunications operators must, on the one hand, adjust their business models to leverage their assets and capabilities, while on the other hand, migrate some of their current services to convergent OTT delivery models, capable of meeting their clients' demands, while reducing their Operational Expenditures (OPEX) and Capital Expenditures (CAPEX).

1.2 Approach & Objectives

Having discussed the key motivation outlines, this section enumerates the chief Thesis' objectives, their importance and how they relate to each other.

The purpose of this Thesis is to improve current OTT multimedia content delivery solutions, taking into consideration the end-to-end delivery chain. The scientific research is focused on a migration use-case of a popular Catch-up TV service from managed IPTV networks, relying on progressive video streaming, to a fully adaptive OTT scenario, starting immediately after content ingestion and ending at the client terminal.

A high level perspective on the goals set forth in this Thesis is shown in Figure 1.2. These objectives are highly intertwined, and may be grouped into three main categories, according to where in the content delivery chain they are applicable.

The first category focuses on characterizing the service at hand and gaining insight on possible optimization opportunities, which is the goal of the *Catch-up TV Consumption Modeling* and *Catch-up TV Demand Forecast* blocks. By thoroughly examining how, when, and what users want, it is possible to create models that may be used to tailor services to their users, on the one hand, and to leverage that knowledge to forecast demand, which is an invaluable tool from a technical and business perspective.

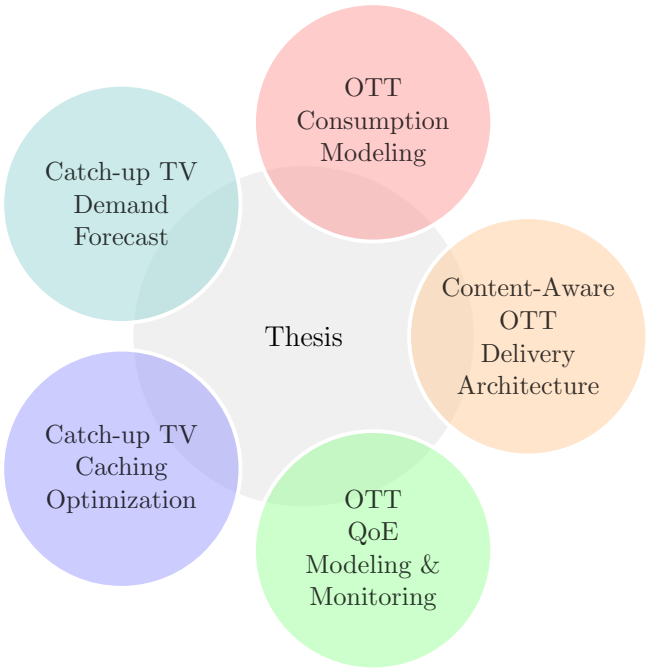


Figure 1.2: Thesis Contribution Objectives.

The next category, comprised of blocks *Catch-up TV Caching Optimization* and *Content-Aware OTT Delivery Architecture*, builds on the previously attained information to improve the service delivery infrastructure and provide scalable and resource-efficient services. The first component explores demand forecasts with the purpose of enhancing a critical aspect of CDNs – caching. The second, embraces a holistic perspective to propose a delivery solution that is able to take advantage of dynamic content characteristics to tune itself and promote efficient resource usage.

The final category targets *OTT QoE Modeling & Monitoring* at the client terminal, given that, to maximize the users' QoE, one must be able to properly measure it. Although several scientific contributions exist on the issue of multimedia QoE, they are lacking in the face of novel dynamic adaptive streaming algorithms used in state-of-the-art OTT.

In spite of the focus on Catch-up TV services, the concepts and methodologies proposed in this Thesis are not limited to this single use-case and may be applicable to other OTT content delivery scenarios, with the necessary context adaptations.

Type	Year	Title	Venue
Conferences	2015	Time-shift services : a taxonomy and techno-business impacts of Catch-up TV	CENTERIS 2015 [25]
	2016	Over-The-Top Catch-up TV Content-Aware Caching	IEEE ISCC 2016 [26]
Journals	2016	Survey of Catch-up TV and Other Time-Shift Services: A Comprehensive Analysis and Taxonomy of Linear and Nonlinear Television	Telecommunication Systems @ Springer [27]
	2016	Catch-up TV Analytics: Statistical Characterization and Consumption Patterns Identification on a Production Service.	Multimedia Systems @ Springer [28]
	2016	Catch-up TV Forecasting: Enabling Next-Generation Over-The-Top Multimedia TV Services.	Submitted: Multimedia Tools and Applications @ Springer [29]
	2017	Content-Aware Over-The-Top Delivery of Catch-up TV Services.	Submitted: Transactions on Multimedia @ IEEE [30]
Book Chapters	2015	QoE Assessment of HTTP Adaptive Video Streaming	Wireless Internet @ Springer [31]

Table 1.1: Publications - Contributions.

1.3 Main Contributions

The main contributions provided by the research work in the scope of this Thesis encompass a broad set of elements within the end-to-end OTT multimedia delivery infrastructure, starting with the characterization of a popular Catch-up TV service, forecasting its demand, dealing with caching optimization, proposing a content-aware OTT delivery architecture, and finishing with QoE modeling and monitoring. A concise description of the accomplished scientific contributions is presented on Table 1.1.

A detailed technological and business oriented evaluation of the increasingly popular category of time-shift TV services is conducted in [25], which is complemented by a worldwide survey in [27] that frames the importance of this class of services, and reinforces the relevance of time-shift TV, and in particular Catch-up TV, in OTT scenarios.

Having established the importance of Catch-up TV services on next generation OTT CDNs, the research work proceeds with a detailed statistical characterization of a popular Catch-up TV service: MEO's "Gravações Automáticas". By leveraging over 22 million request logs of a production service, [28] is able to provide accurate consumption models, extract key insights applicable on content delivery optimization, and gauge potential bandwidth, storage, and caching gains in optimized delivery solutions.

These optimization opportunities are explored in a subsequent work, [29], where the previously generated statistical models are leveraged to evaluate and create demand

forecasts through machine-learning regression algorithms which are shown to enable advanced dynamic resource provisioning systems.

Another practical outcome, enabled by the creation of demand forecasting models, is a novel caching algorithm, MPU, proposed in [26], where it is shown to significantly outperform competing caching alternatives.

These individual research works are put together in an envisioned content-aware OTT delivery architecture for Catch-up TV in [30], where a global framework for improved delivery of Catch-up TV on OTT CDNs is presented, taking advantage not only of the created statistical models, but also of the demand forecasts and proposed caching algorithm, MPU, to provide a delivery system capable of resource-efficient delivery of Catch-up TV while simultaneously improving users' QoE.

The QoE evaluation, which targets the client side of multimedia OTT applications, is essential for benchmarking the performance of content delivery platforms in modern OTT networks that rely on these novel streaming technologies, and is performed according to QoE estimation models and tools developed in [31], which are shown to accurately address the issue of QoE estimation on HAS scenarios.

Intensive development and research work has taken place during the past 5 years in strongly related projects which have had a significant impact in users' OTT experience, on the one hand, but also on academic and industrial fields, namely:

- MEO Gravações Automáticas (Sep 2012 - Present) – Lead development and support of IPTV services and applications;
- GAPOTT, Gravações Automáticas e Publicidade Over-The-Top, QREN SI I&DT 34009/2013 (Mar 2013 - Jul 2015) [32] – Major contributions to all work packages, milestones and deliverables; consortium coordination responsibilities;
- NOTTS, Next-generation Over-The-Top Multimedia Services, Eureka! Celtic Plus C2012/2-4 (Mar 2013 - Mar 2016) [20] – Major contributions to all work packages, milestones and deliverables; Portuguese consortium coordination responsibilities;
- UltraTV, Ecossistema de aplicações para TV UltraHD, Portugal 2020 - 17738/2016 (Mar 2016 - Mar 2018) [21] – Contributions to most work packages, milestones and deliverables;

In addition to the listed contributions, the collaboration with *Altice Labs, SA* throughout the Thesis' work promoted a frequent and intensive knowledge transfer to and from the academia that stimulated innovation, guided research efforts to industrial applications, and ultimately improved the know-how and competitiveness of both parties.

1.4 Outline

The initial overview of the Thesis motivation is presented in the first chapter, with the purpose of framing the relevance of OTT services and their research challenges, stating the key research goals, and identifying relevant scientific contributions.

Chapter 2 provides a detailed examination of key scientific domains, with the purpose of establishing the baseline state-of-the-art. This evaluation starts by taking a comprehensive look into current multimedia delivery infrastructures, their application to telecommunication operators' services, and progresses with an evaluation of the elements that compose modern CDNs, ranging from architectures, to replica server placement, request routing systems, and caching mechanisms, to name a few.

The state-of-the-art evaluation then proceeds to take a look into modern streaming technologies and protocols, which use CDNs as delivery infrastructures, with a particular emphasis on adaptive and scalable streaming algorithms. Expanding on these research topics, a focused examination is conducted on OTT multimedia caching technologies, tailored towards adaptive streaming algorithms. In order to take advantage of the vast amounts of data generated by these modern content delivery systems, data analysis research is surveyed, with a focus on predictive data modeling. An analysis on QoE in the context of OTT video networks is also conducted.

The scientific research work performed in this Thesis is broken down into separate chapters, according to their different goals.

Chapter 3 motivates and characterizes the service under evaluation, Catch-up TV, with the purpose of asserting its relevance as a meaningful migration use-case in the context of OTT CDNs, and assess potential optimization opportunities.

Next, on Chapter 4, the characterization and modeling work is leveraged to present and propose solutions for optimizing OTT CDNs, starting with work on how to measure QoE in these specific scenarios, proceeding with building forecasting models able to anticipate service demand, which is then used to present a novel and improved caching algorithm. The complete research work is then unified in an envisioned content-aware OTT delivery solution architecture.

Finally, Chapter 5 provides a concise summary of this Thesis, the completed tasks, and future research directions.

Chapter 2

State of the Art

2.1 Introduction

To understand the research challenges of Over-The-Top (OTT) delivery networks and the particularities associated with delivering high-performance Catch-up TV services, a deep insight and characterization is required on their usage scenarios and supporting technologies. The introductory Section 2.2 provides a review of standard multimedia delivery infrastructures and the steps involved in content delivery from content preparation up to its playback on client terminal devices. The delivery infrastructure is composed of several key components that are the target of this Thesis research. The section then proceeds with a detailed description of OTT multimedia networks and related concepts, along with an exploration of services delivered by telecommunication operators, with particular emphasis on Catch-up TV services, their relevance on the global Pay-TV landscape, viewership characterization, and impact on market and business models.

Next, on Section 2.3, a structured survey is conducted on the main mechanisms composing modern CDNs, beginning with an overview of their structural architectures and then diving deep into their main responsibilities: monitoring and performance measurement; accounting and billing; management; content ingestion; content distribution and replication; content caching techniques; replica server placement; and request routing.

CDNs are generally agnostic to the content being distributed; however, this characteristic is sub-optimal in multimedia streaming scenarios where content dynamics hinder the performance of existing distribution and caching strategies. Therefore, Section 2.4 conducts an analysis on multimedia streaming protocols, starting with the well established progressive streaming, and moving on to adaptive and scalable streaming mechanisms, in the context of live and on-demand delivery. Due the specificities of OTT multimedia content delivery, caching algorithms utilized by specialized multimedia CDN play an important role on the overall performance of the delivery solution; therefore, these specific algorithms are surveyed on Section 2.5.

A trending topic in most commercial services with exposure to end-users is that of Quality-of-Experience (QoE). In the context of OTT multimedia streaming, this is also a relevant issue, as it represents the user-perceived service quality, which, ultimately

might be more important for the success of a commercial service than other individual network oriented metrics, such as those typically encompassed in QoS parameters. This issue is addressed in detail on Section 2.6, which comprises an overview of chief QoE concepts and their applicability to OTT multimedia networks.

Given the huge amounts of data generated by modern large-scale delivery services, or *Big-Data*, the problem of data processing with the goal of extracting useful information that might be used by the services, in a feedback loop, is of crucial importance. This information, or knowledge, must be extracted through data-mining techniques, relying on machine learning, with the purpose of building predictive models that can shed insight on the expected service behavior under varying circumstances. Due to the relevance of this topic and its applicability as an optimization mechanism to OTT multimedia networks, a state-of-the-art review on this subject is conducted on Section 2.7.

Figure 2.1 presents a mind map summarizing the addressed key topics.

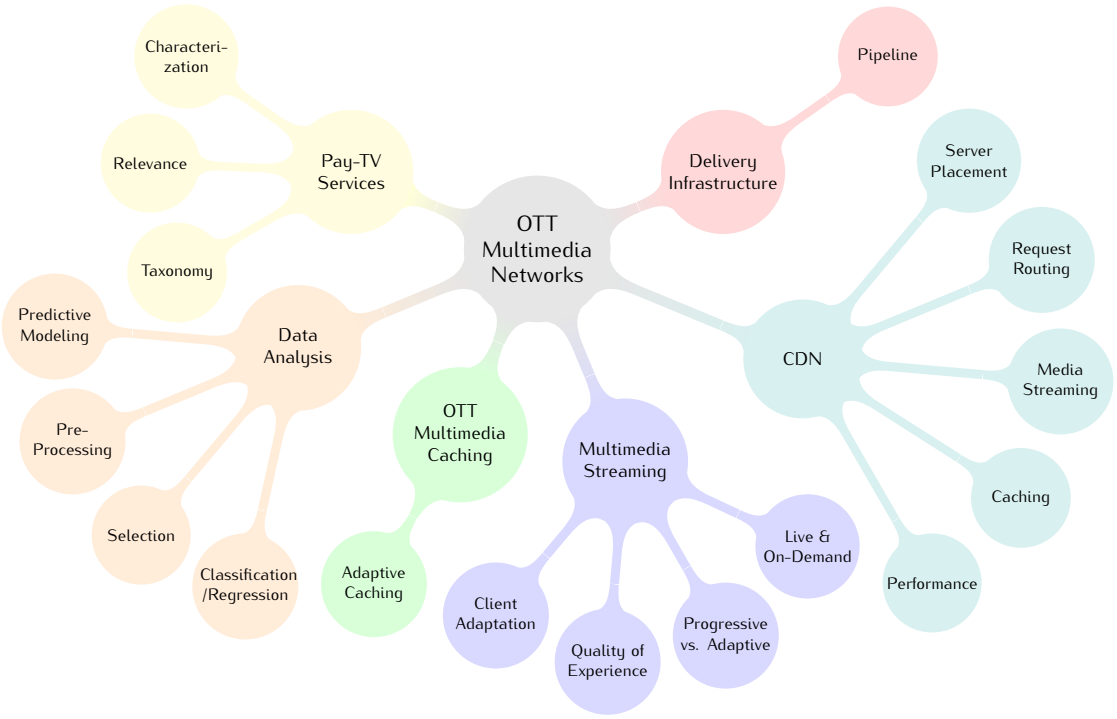


Figure 2.1: Mind Map of the Main Research Topics Addressed.

2.2 OTT Multimedia Networks & Services

The Internet evolved from a simple messaging system [33] built on the shoulders of Internet Protocol (IP) and the end-to-end argument [34] to become one of the most complex systems in operation, both in number of protocols supported and sheer scale.

Over-The-Top multimedia networks represent a growing class of media delivery technologies, whose distinctive characteristic is the unmanaged (or open) delivery, without network-supported stream control. The classification of *closed* or *open* networks depends on who controls the network [35].

In a closed (or managed) network, the delivery is performed with the involvement of the ISP, which ensures predetermined QoS features. This is the type of service that is provided on commercial IPTV platforms such as Ericsson's Mediaroom [36].

On the other hand, in an open and *uncontrolled* network the delivery takes place without any interference or quality guarantees of the ISPs supporting the delivery, which takes place as if it were any regular Internet content. Because the ISPs' networks are being used to provide a service from a third-party, which typically uses their network infrastructure for free, this type of delivery is called Over-The-Top.

The characteristics of OTT networks enable services to be delivered to the whole Internet, without any capital or operational expenditures on the network infrastructure itself, which are supported by the intermediate ISPs. However, there are some drawbacks to these services: because the networks they rely on to operate are not controlled, no quality guarantees may be ensured, and OTT providers depend entirely on the supporting best-effort network.

This fact raises multiple issues as far as the users' QoE is concerned. The high-QoE goal requires scalable, reliable, and adaptive services, which must be able to infer the environment conditions in *quasi*-realtime in order to provide the users' with the best possible experience at a given point in time. In the context of video delivery, a good experience is correlated with metrics such as low-buffering times, no video freezes or macro-blocks, a video resolution adequate to the viewing-device's screen and, in live events, low end-to-end delay, to name a few.

To frame and provide a good understanding on the services under consideration and how they are delivered, this section begins by illustrating and discussing each step of the typical content delivery pipeline, and proceeds with a detailed evaluation of a widespread class of OTT multimedia services: that of telecommunication operators, with emphasis on Catch-up TV.

2.2.1 Content Delivery Pipeline

As the requirements for bandwidth and demand for ever-richer applications grew, so did the need for systems and architectures able to withstand it in a scalable fashion. In the multimedia delivery context, one that requires huge amounts of bandwidth, a *de facto* standard delivery pipeline naturally arose and was established to represent the multi-party nature of the multimedia life-cycle, from content creation to consumption.

In order to understand the steps involved in making multimedia content available to users for mass consumption, a content delivery pipeline is exhibited in Figure 2.2, illustrating the various parts and components of a modern delivery process.

This diagram is by no means exhaustive, but provides a good macro perspective on the delivery process.

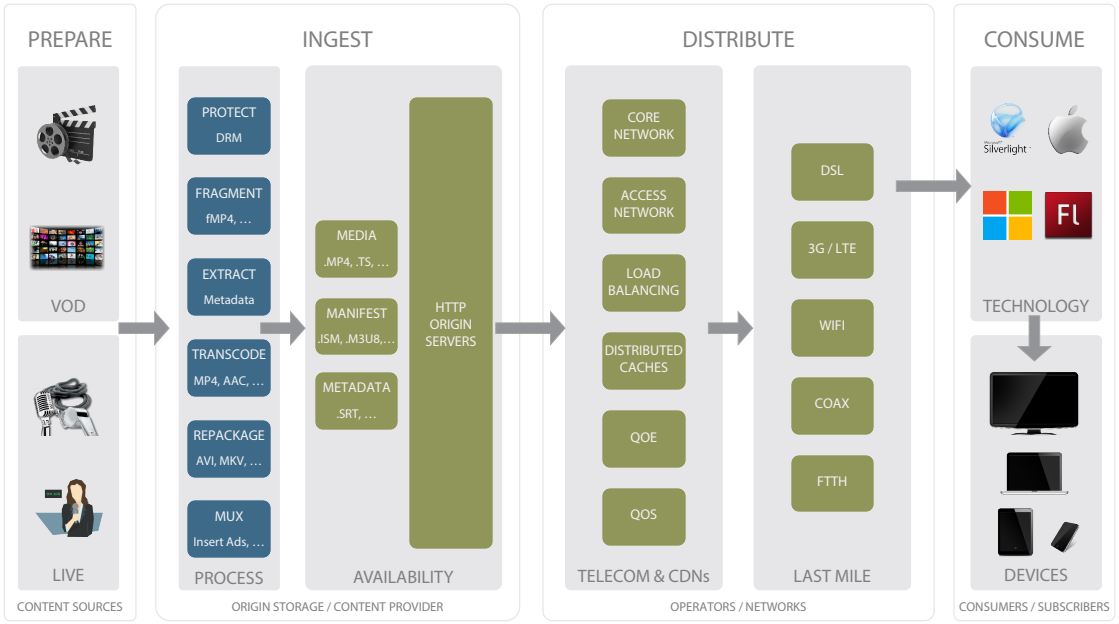


Figure 2.2: Content Delivery Pipeline Example.

Content Preparation

The preparation portion of the delivery pipeline performs the actual acquisition of the content in some multimedia format from a live broadcast or VoD media (DVD, etc), transforms the original content into the content that will be displayed to the user – e.g. ad insertion in live broadcasts –, encodes it into a consumer-friendly format, such as H.264 or the more modern H.265 codec, and if necessary applies content protection through Digital Rights Management (DRM) mechanisms. At the end of this step, the content is ready to be used as a master source for distribution.

Content Ingest

Ingestion is the act of taking a content that is ready for distribution and making it available to the delivery network by placing it in a top level content source. The content is copied into a set (for redundancy and scalability) of so-called *Origin* servers which hold master copies of the multimedia content and expose it to distribution networks.

Content Distribution

The actual content distribution encompasses two main components: the network elements with their associated access technologies (such as FTTH, Digital Subscriber Line (DSL), ...); and the content distribution servers inside the said network, that may have different levels of complexity.

This component is also responsible for monitoring the provided QoS and QoE, although situations exist where this monitoring is performed by the media servers or the client devices (e.g. in Microsoft Smooth Streaming [37] monitoring is performed by the client as well). The content distribution servers may range from full-blown CDNs, to simple load-balancers ensuring that the available origin servers get similar load shares.

Consumption

The final step in the content delivery pipeline is the actual media consumption by the client device. How it's performed depends heavily on the device, its Operating Systems (OSs), and hardware specifications, such as decoding abilities, screen size, etc. Given the heterogeneity of devices, operating systems, and technologies on the market, this step usually represents a formidable challenge.

2.2.2 OTT Multimedia Services of Telecommunication Operators

Even though OTT multimedia networks are, by definition, not constrained to telecommunication operators, and key market players exist with "pure-OTT" business models (Google, YouTube, Facebook, ...), telecommunication operators play a very important role in shaping the OTT industry and its services. Their Pay-TV offerings provide rich and interactive services that are widely used across the world; thus, telecommunication operators often provide the push for the massification of new services.

Historically, telecommunication operators have relied on managed networks to deploy their services; however, with the advent of OTT supported services which provide strong competition, they are also moving towards OTT-based distribution systems.

In virtue of this shift to OTT services that complement the managed ones, it is desirable to understand their nature so that their details might be taken into consideration when discussing CDNs (Section 2.3), streaming protocols (Sections 2.4 and 2.5, and QoE (Section 2.6).

Telecommunication operators may provide OTT services as a multi-screen extension of their Pay-TV services to give users the choice of accessing the content and services they want, and pay for, in a wide range of client devices, instead of being constrained to a location (home) or a device (TV). On the other hand, OTT services enable operators to provide standalone services on otherwise unreachable situations, as the requirement for managed networks is no longer in place, thus broadening consumer choice – i.e. a given consumer might have Internet from one provider and a linear TV subscription from another.

Linear TV

Linear TV, i.e. "regular TV broadcast" obeying to a predetermined program line-up was considered for decades as the traditional and more popular way of watching TV programs. These were the times where delivery networks were monopolized by public and private operators carrying their own TV programs. Nowadays, this is still the dominant way of watching TV from national free-to-air TV services and major Pay-TV Operators like BT in England; NET in Brazil; Time-Warner in the USA and MEO in Portugal, although customers are moving to other services [38].

Time-shift TV

Time-shift TV relates to the visualization of deferred TV content, i.e. linear-TV content that is recorded to be watched later (from seconds up to several days), using one of the following services:

1. *Pause TV* is the simplest type of time-shift service, allowing users to pause the television program they are currently watching - from a few seconds to several minutes or even hours. Users can resume the TV broadcast when they want, continuing to watch where they left off; skip a particular segment; or eventually catch up to the linear broadcast.
2. *Start-over TV* enables users to restart programs that have already started and, eventually, programs that already finished. The amount of time that is possible to rewind varies from operator to operator ranging from some minutes up to 36 hours. The number of TV channels supporting this feature is also a decision of the TV operator.
3. *PVR* stands for Personal Video Recorder. In this type of service the recordings are subject to the user action, i.e., they only occur if the user proactively schedules a TV program or a series to be recorded, or if he decides to start recording a program that is being watched. The behavior of the service is much the same as the one of a VCR (Video Cassette Recorder); however, with a much higher storage capacity and nonlinear access. The user can start watching a recording whenever he wants, even if the program is still being recorded.
4. *Catch-up TV* is the most advanced time-shift service, relying on an automated process of "Live to VoD" [39] (offered by companies like Alcatel-Lucent [40]) or on a more restricted process-based editorial control. With this service, TV operators offer recorded content of the previous days, on a bouquet up to hundreds of TV channels. The time window of the recordings ranges from a couple of hours up to 30 days, and the number of recorded TV channels varies from operator to operator, according to technical, legal, and business constraints. Using this service, users can really, and very easily, catch-up TV programs that have been missed or that they explicitly decided to watch later – e.g. watching the news after preparing dinner.

Video-on-Demand (VoD)

VoD refers to services where users need to pay to watch a specific content through one of the following ways:

1. Transaction VoD (T-VoD) is the most typical version of the service, where customers need to pay a given amount of money each time they want to watch a content from the VoD catalog. The rental time is usually of 24 or 48 hours, during which the customer may watch the content several times.
2. Electronic Sell Through VoD (EST-VoD) is a version of the VoD service involving the payment of a one-time fee enabling customers to access the purchased content without restrictions, usually on a specific platform. Although this method of VoD is more frequent in OTT providers like Apple iTunes and Amazon Instant Video, it also being offered by traditional Pay-TV operators, like Verizon's FiOS TV.
3. Subscription VoD (S-VoD) corresponds to the business model also adopted by OTT providers like Netflix, whereby customers pay a monthly fee that allows them to watch whatever they want from the provider catalog for an unlimited number of times. However, like the EST-VoD version, it is no longer an exclusive option of these providers, since Pay-TV operators are also offering S-VoD. A simple example is the Disney VoD service offered by several Pay-TV operators like AT&T, Cablevision or Comcast.

2.2.3 Impact of Catch-up TV Services

Several studies indicate a revolution in the television ecosystem due to the introduction of manual and automatic recordings, recommendation and retrieval technologies for television content. Among non-linear IPTV services, Catch-up TV distinguishes itself as the most popular one, even surpassing the popularity of "classical" VoD services such as T-VoDs or EST-VoD [41, 42].

Large scale delivery of Catch-up TV represents one of the biggest challenges of Pay-TV, mostly due to two reasons: first, the content must be streamed in unicast to each client, with dedicated connections per user; second, Catch-up TV content demand is several orders of magnitude larger than that of traditional Video-on-Demand (VoD) content [42].

To lessen the impact of unicast traffic, [43] and [44] suggest the use of decentralized delivery solutions whenever possible, with subsequent studies in [45] for cable television networks, and in [46, 47, 48] for IPTV services.

The fact that Catch-up TV is data-intensive is challenging, as it is usually provided as a supplement to Pay-TV subscriptions with no added cost. The network impact of Catch-up TV is expected to keep growing with its popularity, which has been one of the main drivers of an increase in the average time spent by users watching TV [38]. To keep-up with a growing demand, IPTV operators are turning to OTT delivery solutions which do not require investments on managed IPTV infrastructure, and increase the reach of services that may have been previously limited to certain geographic areas.

However, this move requires overcoming several challenges. Given the different re-

quirements of OTT delivery, when compared to that of managed networks, a detailed service understanding is required to properly decide on OTT CDN architectures, plan the physical and logical location of clusters and replica servers, tune caching algorithms, select optimal request routing mechanisms, and estimate computational, network and storage requirements, to name a few.

From an operator's viewpoint, a thorough service comprehension fosters savings on both CAPEX and OPEX. As an example, CAPEX may be reduced by investing on less extra capacity, because the exact service requirements are known and the delivery system is optimized to meet them, which also contributes to reducing the OPEX.

2.2.4 Usage of Catch-up TV Services

To design OTT Catch-up TV delivery systems capable of operating efficiently, while simultaneously improving users' QoE, it is also essential to understand how the service is effectively used by the clients.

From a behavioral perspective, [49] presents a descriptive and inferential statistical analysis on viewing practices (time-shifted, online and mobile), based on data collected over a six-month period in 2010-2011. The authors consider that the popular time-shift services do not alter the traditional conceptualization of television as a broadcast medium; however, they do not make a clear differentiation between the diverse time-shift services (Pause-TV, Start-over TV, Personal Video Recorder (PVR) and Catch-up TV). Online viewing, considered an emerging mode that blurs the boundary between television and new media, is seen by the authors as comprising P2P, Bit-Torrent and video streaming from network TV station sites or dedicated services (e.g. Netflix). As for the motivation that drives respondents to watch content on their computers instead of their TV sets, the reason that stands out is the lack of content availability on broadcast television (42.5%). Finally, they present mobile viewing, mostly through dedicated applications, as the most recent consequence of digital convergence. Despite the potential evolutions registered from 2010-11 until now, the paper gives worthy insights about the differences across three key demographic variables: gender, age, and region of residence.

A recent paper, [50], performs an interesting comparison of broadcast TV viewing behaviors with several nonlinear services (Catch-up TV, VoD streaming services, content recording and downloading). They found that TV series and movies are mostly watched through nonlinear services, and also corroborated that people's attention to content is more focused when nonlinear services are at stake, whereas with regular broadcasts (news, talk shows and other "lighter" television genres) the adoption of multitasking behavior is more frequent. Finally, the authors also illustrate that the hassle of dealing with the several fragmented services, with different qualities, prices, and technological issues can make it hard for users to watch TV the way they want. This merging of household media devices and delivery systems was already pointed by Jenkins when he referred to the Black Box Fallacy [43].

These works are consistent with other research, such as [51], which claims that online content consumption is more concentrated in time and quantity than offline viewing, contradicting Catch-up TV's long tail hypothesis. The authors state that 69% of the

videos have the same success online and offline; 16% of the videos are not successful in any platform and only 15% benefit from being available online. The temporality of replay TV consumption is very close to live broadcasting, thus softening rather than breaking the synchrony of traditional TV. The largest consumption of online videos happens in the first 3 days of their appearing, with 58% of the total views. The study is limited to 11,682 videos available on a 5-month window and to 7 TV channels. Similar results were attained by [52], which adds that users overwhelmingly prefer serialized content.

In addition to the research works focused on Catch-up TV, other measurement studies exist that characterize and model key aspects of IPTV services such as linear/live TV, and T-VoD services. In the work by Cha *et al.* [53] the users' live TV channel changing behavior is exhaustively analyzed. The work's chief conclusions indicate that most channel switching events happen within 10 seconds, suggesting that users' have a very volatile focus. Other key findings pertain to the channels' popularity, which is found to change with the time of day, and to daily viewing patterns, which vary with the channels' genres. Gopalakrishnan *et al.* [54] leverage traces across a 2 year period from a large-scale IPTV service to provide models for the video request *arrival process* and *stream control* of a T-VoD service. A detailed characterization shows that VoD assets may be grouped into 5 separate clusters of video lengths, that the video popularity distribution follows an approximate Zipf distribution, and that a strong popularity drop-off exists as the content ages, showing that a content's recency influences its popularity.

2.2.5 Conclusion

OTT multimedia networks are significantly different than traditional managed networks, and provide an opportunity for creative new usages of the network infrastructure, as can be observed by novel OTT service providers such as Google, YouTube, Facebook, or Netflix. However, in a world still dominated by telecommunication operators, their services continue to be very relevant and are also making the shift to OTT oriented delivery solutions. Catch-up TV services emerge as a key migration challenge to address, due to their massive popularity and infrastructural demands.

To better understand how this Thesis addresses these challenges, an in-depth overview of the technologies and issues present on OTT delivery needs to be performed, and is thus the focus of the following sections.

2.3 Content Delivery Networks (CDNs)

CDNs emerged in 1998 [55, 56, 57] to become an essential piece of modern delivery infrastructures. They were developed to overcome the exponential growth of the Web in terms of demand for bandwidth, content, and services and strive to reduce the load on origin web servers, and to increase the users' perceived QoS. Their use has a tremendous impact on the scalability of these servers and helps with providing users with the information they request in a timely manner, especially in "flash crowd" or "Slashdot" [58] scenarios where demand for particular content peaks well beyond reasonable resource provisioning.

In a CDN, the content is replicated over a number of servers performing collaborative tasks, whose purpose is to transparently and efficiently deliver content to the end users from an original Web server, the *origin* server. They facilitate request redirection to redirect the users to the content that is closer to them, content outsourcing to seamlessly add scalability to existing web servers, and content management services to provide accounting, monitor usage and generate reports, among others.

In the context of a CDN, the *content* refers to the data being managed and its associated metadata, i.e. additional information about the data that enables complex tasks and rich environments such as context-awareness, discovery, and indexing [59].

Example of applications of CDN technologies may be found all over the Internet:

- *Academic Institutions* : Codeen [60], Coral [61], Flash Crowds Alleviation Network (FCAN) [62];
- *Network Operators* : AT&T, Telefonica, Telus, British Telecom;
- *Social Networks* : Facebook, MySpace, Twitter;
- *Online Retailers* : Amazon, eBay, NewEgg;
- *Media Providers* : YouTube, Netflix, Hulu, iTunes;
- *Service Providers* : Akamai, CloudFlare, Amazon CloudFront.

The ensuing subsections will discuss CDNs in detail, illustrate the reasons behind their massive usage, and the research challenges they pose.

2.3.1 Structural Architecture

CDNs are composed of multiple servers, sometimes called the *replica* or *surrogate* servers that acquire data from the *origin* servers, which are the actual source of the data. The replica servers are interconnected and store copies of the origin servers' content so that they can serve content themselves, reducing the load of the origins.

The issue of how to build a CDN is usually solved using *overlay* or *network* approaches, although hybrid solutions are common [63]. The overlay approach establishes application-specific connections and caches on top of the existing network infrastructure and creates virtual interconnections between the network's nodes. In this approach, the network elements such as switches and routers do not play an active part in the content delivery and management process, other than providing basic connectivity and

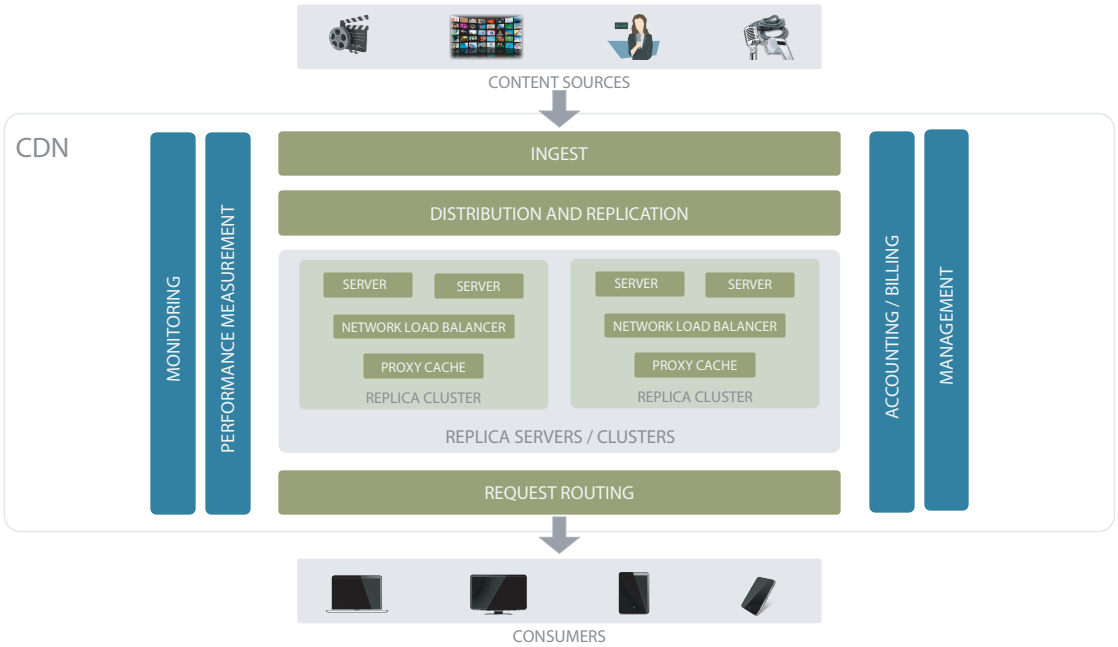


Figure 2.3: Typical Content Delivery Network Architecture.

any agreed-upon QoS Service Level Agreements (SLAs) for the CDN nodes. This is a widely used approach in commercial CDNs such as Akamai [64] and Limelight [65], as the independence from the underlying network components warrants flexibility not only in terms of the services that can be provided, but also in node deployment.

In contrast, the network approach relies on network elements such as switches, load balancers, and routers to forward requests to local caches and/or application servers according to previously established policies. This is a less flexible form of CDNs that is usually heavily optimized for serving specific content types, and that is often used as a complement to overlay approaches in server farms, i.e. a server farm may internally use a network-based approach for CDN despite being part of a larger overlay network.

Depending on how the CDN is devised, multiple protocols may be used in the interaction between the different replica servers, such as Cache Array Routing Protocol (CARP) [66], or Hypertext Caching Protocol (HTCP) [67]. Apart from these common protocols, each vendor / designer of CDNs usually implements its own communication or interaction protocols, such as Railgun from CloudFlare [68].

In order to get the most performance out of a CDN, it is usual to create platforms that are either tailored or adaptable according to the content they are serving. A good example is the multimedia streaming services of Akamai HD [64], which are optimized for streaming, and the application services of CloudFlare [68], which are optimized for serving dynamic applications.

No single CDN solution is able address every possible service in an optimized manner without adaptation, i.e. a single optimal and universal solution for CDNs does not exist.

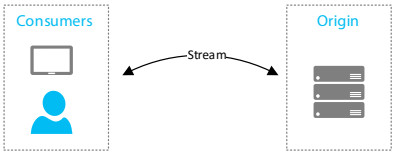


Figure 2.4: Centralized OTT Delivery.

A CDN is in itself a complex network that is expected to perform a multitude of tasks, which are usually subdivided into three main functional components: delivery and management; request routing; and performance measurement, billing and accounting. Apart from billing and accounting, each responsibility will be individually considered and scrutinized. Figure 2.3 provides an all-encompassing conceptual perspective of the main components of a CDN.

An ideal OTT multimedia CDN is able to deliver content without significant delay and scale-out in order to grow capacity as needed with the simple addition of servers. A state-of-the-art global CDNs is required to sustain bandwidths in the order of Tbps.

Content Distribution Architectures

In order to understand which content distribution architectures are scalable and viable in a large scale OTT delivery infrastructure, an evaluation must be performed on the possible ones, to identify each solutions' strengths and weaknesses [69, 70].

Centralized Content Delivery The centralized approach to OTT delivery is the simplest one, where the clients directly target the origin servers without any intermediate edge server, as depicted on Figure 2.4, and a unicast stream is created directly between a given origin server and a consumer device.

Because consumers target the origin servers directly, this approach provides the lowest delivery delay when streaming live content, and may be cost effective for a few users - small being defined by the maximum number of users that the origin cluster is able to serve simultaneously.

However, this fully centralized approach presents several issues. Firstly, there are security constraints, usually imposed by content providers which forbid users from having direct access to origin servers.

Secondly, this approach does not scale properly with geographically distributed consumers, as users that are further away from the origin cluster will experience increased access delay to the content, which is even more problematic if the streaming session is being conducted using Transmission Control Protocol (TCP), as this protocol is known to underutilize links when faced with long network delays, especially in high-bandwidth networks - i.e. Long Fat Networks (LFNs) [71]. These characteristics will naturally lead to user frustration and reduced QoE.

Thirdly, a centralized approach, without carefully planned content replication requires a large amount of core network and transit traffic, which is particularly expensive in multimedia streaming scenarios, known to have high bandwidth requirements.

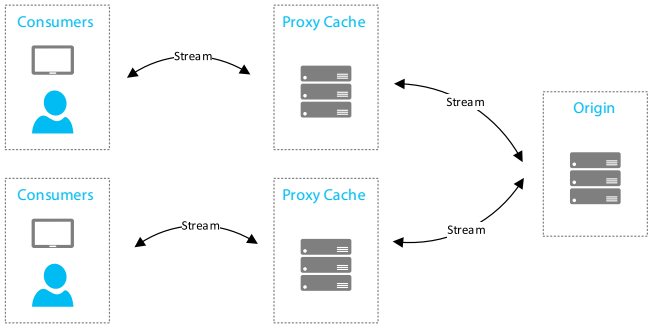


Figure 2.5: Proxy Cache OTT Delivery.

Proxy-Caching An alternative approach to the centralized solution is the proxy cache architecture, where intermediate, or proxy, nodes communicate directly with the consumers, acquire the content from the origin servers on their behalf, and cache it. This architecture is illustrated on Figure 2.5.

This approach presents several benefits when compared with the centralized one. Content security is increased due to the indirect access to the origin server, which may be put in a private network, as long as proxy caches have access to it. Scalability is also increased, by means of caching and geographical distribution of proxy caches, which also provided the added benefit of improvements in users' QoE due to reduced access latency to the servers in the likely event of a cache hit. Bandwidth costs are reduced through savings in core and transit network traffic.

In spite of these advantages, a proxy cache solution has potential drawbacks, due to two main factors: increased management and deployment complexity; and increased end-to-end delay in the event of a cache miss, which may have a significant impact in the case of live content streaming. However, the benefits of this approach often outweigh its drawbacks.

Peer-to-Peer (P2P) A P2P approach to distributing OTT content is another possibility where both the proxy caches and the consumers may communicate with each other to locate and acquire content, as exhibited in Figure 2.6. P2P takes advantage of the uplink capacity of users' and proxy caches' connections to lessen the bandwidth burden on the origin servers. An example of a widely used fully P2P streaming service is Popcorn Time [72].

In spite of its advantages in terms of utilizing the available upstream bandwidth, P2P streaming presents several challenges that prevent it from being widely deployed on OTT content delivery networks. One of the issues has to do with the startup delay of new streaming session, as locating and acquiring data from peers takes longer than streaming directly from an origin server or proxy cache. Another issue has to do with playback lag in live streaming, as the P2P approach leads to additional delays.

There are also issues with traffic engineering, given that P2P protocols have not been designed to be ISP-friendly. Finally, as with other highly distributed systems, the

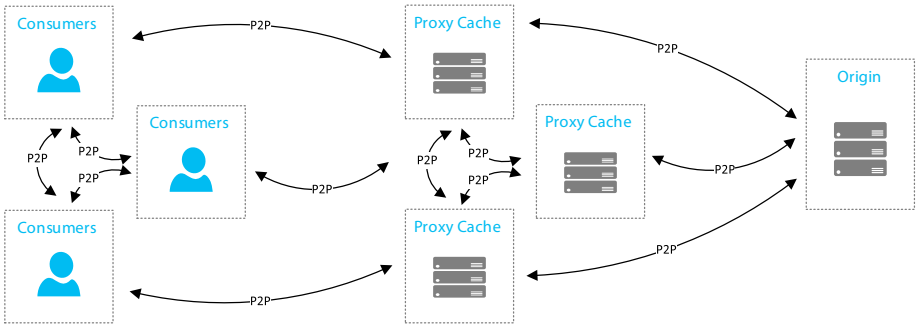


Figure 2.6: P2P OTT Delivery.

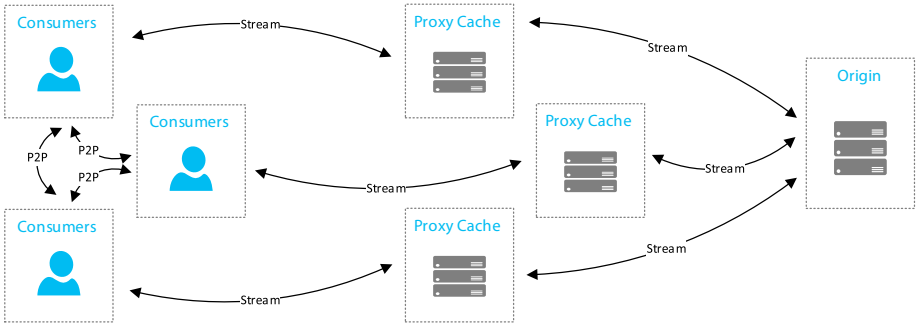


Figure 2.7: Hybrid OTT Delivery.

complexity of deploying, configuring, and managing a fully distributed content delivery network is high, which may be a deterrent for subscription-based streaming service providers that have a responsibility of providing a service with predictable quality to their clients.

Hybrid Delivery Another possible architecture for an OTT streaming service relies on hybrid delivery, and combines P2P at the clients with the previously discussed proxy cache approach. The diagram of this solution is presented in Figure 2.7.

Comparing this approach with the full P2P one, several advantages are apparent. Firstly, because the proxy caches may be used to stream content directly to the consumers, like in the simple proxy-cache approach, the additional startup delay caused by the P2P overhead may be mitigated. Secondly, due to peering, bandwidth and load are saved on the proxy caches, thus allowing the solution to better handle flash crowd events, or very popular content, which will be more likely to be available at peers.

The low latency requirement for live content may prevent P2P approaches to live streaming; however, on-demand content may benefit from it.

The downsides are similar to those of full P2P approaches, in that extra complexity is required in the implementation, deployment and management of a hybrid delivery network. Additionally, some terminals, such as Set-Top-Boxes (STBs), may not support P2P at all.

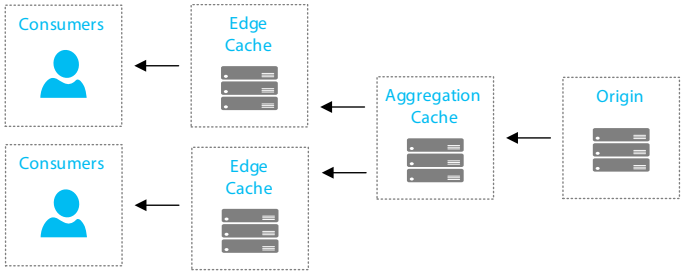


Figure 2.8: Example of 2 Tier Caching Architecture.

N-Tier Caching A design decision that has a high impact on the performance of a CDN is the amount and disposition of caching layers, as well as the storage space available at each layer [73].

The simplest approach to caching within an OTT CDN is to place a single proxy-caching layer at the edge servers, which is responsible for fetching content on behalf of the client and storing local copies according to predefined caching polices, as previously shown in Figure 2.5.

In more elaborate approaches, it is possible to add supplementary caching layers, which take the name of *aggregation caches*, as opposed to the client-facing *edge caches*. Figure 2.8 depicts an example of a 2-tier caching solution.

There are several advantages of having aggregation caches on top of the edge caches. When a user moves from one edge cache to the other (as a result of mobile base station change for example), content previously cached on the edge cache and also on the aggregation cache, does not need to be re-requested from the origin server, and may be served directly from the aggregation cache to the new edge cache.

In the event of an edge cache failure, due to a server failure for instance, the aggregation cache is also useful given that, in order to rebuild the cache of backup edge caches, there is no need to target the origin server, provided the content is present at the aggregation layer.

An aggregation cache is also useful when a CDN has a high geographical diversity and potentially large access delays to the origin server. Using as an example the USA and its 50 states, an aggregation cache could be placed on each state, while edge caches would be installed on each main city.

In these scenarios, a trade-off that must be considered is the cost of adding aggregation caches instead of investing in larger edge caches. Finally, due to the introduction of an additional element in the distribution chain, the end-to-end delay is expected to increase slightly for non-cached items.

Web Servers and Content Caching Technologies

Having detailed the most commonly employed structural CDN architectures, this subsection details commercial implementations of one of their key components – the web servers – along with their chief caching features.

The advances in the field of hypertext systems in the mid 80's contributed as a starting point to the modern web-servers available today. According to Netcraft's web server survey [74], Apache (37.00%), Microsoft's Internet Information Services (IIS) (30.40%) and Nginx (16.65%) are the most popular web servers on the Internet. These web servers are modular in nature and support caching add-ons capable of turning the web servers into powerful caching systems implementing policies able to take into consideration multiple content types, including live or video-on-demand video, static content, and dynamic web pages, to name a few.

Popular solutions for proxy-caching modules and systems include Apache Traffic Server (ATS) [75], Microsoft's IIS with Application Request Routing (ARR) [76], Nginx [77], Varnish [78], and Squid [79]. They may be combined to optimize the cache performance of a delivery system, e.g. Apache web server with an Nginx proxy-cache front-end). Microsoft's IIS ARR [76] is an extension for the IIS web server supporting rule-based routing, load balancing and distributed caching.

As for open source solutions, ATS [75] is a modular high-performance proxy server originally developed by Inktomi and recently open-sourced by Yahoo!. Just like IIS, it supports reverse and forward proxy, caching solutions, and also request routing, filtering, and load balancing. Squid [79] is an iconic proxy-cache solution, initially released in 1996 as a fork of the *Harvest* research project [80], that is frequently used in academic works.

Nginx [77] is another popular webserver with proxy cache capabilities, originally developed with the goal of addressing the underperformance and scalability issues associated with Apache server. Nginx has a strong focus on high performance and low memory usage when the subject is to serve dynamic HTTP content. With respect to Varnish [78], it is an HTTP accelerator cache solution designed mainly to support content-heavy dynamic web sites focused exclusively on HTTP content. Varnish stores the cached information in virtual memory and leaves the task of deciding which content to cache in charge of the OS. In addition, Varnish works by handling each client connection in a separate worker thread; when the limit of active worker is reached; incoming connections are placed in an overflow queue.

Table 2.1 provides a comparison of features of the evaluated proxy cache solutions.

Supported Features	**ATS**	**Nginx**	**Varnish**	**IIS with ARR**	**Squid**
Reverse Proxy	Yes	Yes	Yes	Yes	Yes
Forward Proxy	Yes	No	No	Yes	Yes
Transparent Proxy	Yes	No	No	Yes	Yes
Plugin APIs	Yes	Yes	Yes	Yes	Yes
Cache	Yes	Yes	Yes	Yes	Yes
Edge Side Includes (ESI)	Yes	No	Partial	Yes	Yes
Internet Cache Protocol (ICP)	Yes	No	No	No	Yes
Secure Sockets Layer (SSL)	Yes	Yes	No	Yes	Yes
SPDY	Yes	Yes	No	No	No

Table 2.1: Comparison of Proxy Cache Solutions.

2.3.2 Content Delivery and Management System

The content and delivery management system is at the core of any CDN. Its responsibilities encompass the replica servers' physical and virtual placement, content selection and actual delivery, the caching organization techniques, and content outsourcing, i.e. how the content is acquired into the CDN replicas.

Replica Server Placement

As the replica servers hold the actual data of a CDN system, their physical and virtual placement plays a very important role on the overall performance of the CDN. Their placement must be carefully planned so that they can be as close as possible to the clients. They usually do not need to be very close to the origin Web servers, as there are typically no significant link bottlenecks between server farms and the Internet; however, if far enough, the network latency might impact their performance.

From a physical location standpoint, the issue of replica server placement may be thought as an instance of the more general problem of placing N servers in M possible locations, such as the facility location of minimum *k-median* problems [81] or *k-Hierarchically well-Separated Trees (k-HST)* [82].

These theoretical approaches to replica server placement define a center placement problem using graph theory, where the goal is to identify N cluster centers relying on some measure of "distance" between the centers and the nodes of the graph, i.e., the clients. In the context of CDNs this distance may combine factors such as available bandwidth, delay, or even costs for transit links.

Similarly to *k-means clustering* [83], cluster cells are created as an outcome of the placement algorithm. This problem is known to be Non-deterministic Polynomial-time (NP) hard; hence, alternative heuristics-based node placement algorithms are commonly used. Examples include *Greedy replica placement* [84] and *Topology-informed placement strategies* [85] that leverage existing information regarding the CDNs such as workload patterns and network topology to provide good-enough solutions at a much lower computational cost. The Greedy algorithm [84] is iterative and chooses the location, at each step, that minimizes the cost for clients, out of the available ones. As for Hot Spot [84], a rank is made regarding the load generated by clients in the vicinity of a possible location, and top N locations are chosen as "hotspots" and consequently the target locations for the N servers. Other derivations of greedy algorithms exist and are discussed in [86].

Some authors proposed dynamic replica placement algorithms [87] that take into consideration QoS requirement for the CDN as well as the maximum capacity of each server/server-farm location; therefore, providing a better approximation of theoretical models. [88] provides a good evaluation methodology on the performance of several heuristic algorithms according to the specific requirements of a given CDN.

Apart from node placement algorithms, there is also the issue of how many replica servers to deploy. This question will vary greatly with the deployment scenario, that may either be confined to a *single-ISP* or *multiple-ISPs* [56].

In single-ISP deployment situations, the size and span of the ISP usually limits the number and locations where the replica servers may be placed, and the choice usually falls on major cities and relies on large capacity servers or server farms. This has the main drawback of potentially not having servers nearby clients that need them. If the ISP has a global coverage, this problem is somewhat mitigated.

An alternative is to use multiple ISPs that can provide a broader choice for node placement, and number of nodes that can be placed. This is the approach followed by large international CDNs such as Akamai that have thousands of servers [64, 89].

Deciding on how many ISPs to use and how many servers to deploy has an impact on the cost and complexity of the CDN. Deployments with an excessive amount of servers may exhibit both poor resource utilization and lackluster performance [57], whereas using fewer than needed servers will also have a significant impact on the performance, albeit due to the excessively high load. A balance must be struck between the expected utilization of resources and the number of replica servers [55].

Server Selection and Content Outsourcing

The next step, after properly placing the replica servers on a CDN, is to decide on how content should be replicated to the replica servers. This is commonly known as *content outsourcing*, and has been the target of vast research.

Traditionally, three main categories for content outsourcing have been established:

- *Cooperative push-based*: content replication based on pre-fetching with cooperation from replica servers;
- *Non-cooperative pull-based*: content replication similar to traditional caches without pre-fetching and cooperation;
- *Cooperative pull-based*: an evolution of the non-cooperative pull-based where the replica servers cooperate with each other in the event of a cache miss;

As for the cooperative push-based approach, its aim is to proactively prefetch content that is expected to be requested by clients and replicate it to surrogates according to some predefined rules or cost functions. The content is pushed from the origin servers to the CDN replicas in a cooperative manner, and information is maintained regarding what content is on what servers, allowing easier request redirection. It is clear then that this problem shows great similarities to the replica placement problem; hence, being NP-hard [90] and requiring heuristics for feasible solutions. Greedy algorithms have been shown to provide a better performance than other heuristics [91, 90].

This approach is traditionally not used on commercial networks given that proper content placement algorithms require knowledge about the Web clients and their demands, which is data that is not commonly available for CDN providers [92, 93].

Regarding the non-cooperative pull-based approach, this is the simplest form of content placement on the surrogates. If a client requests a content that is not on the surrogate, a cache miss is triggered, and the surrogate fetches the content from an origin web server. There is no explicit coordination, or cooperation, between either the sur-

rogates or the origin web servers, the content replication is purely client driven. In its simplicity lies the key for successful deployments on popular CDNs such as Akamai or Mirror Image. The drawback is the natural lack of optimization in the server selected to serve the request [94].

In the final approach, cooperative pull-based, which is being used in academic networks such as Coral [61], the content is also not prefetched, but upon a cache miss the surrogates cooperate in order to find neighboring servers that can accommodate the request and avoid requests to the origin servers. This approach typically draws concepts and algorithms from P2P technologies such as Distributed Hash Tables (DHTs) to foster cooperation between surrogates [95], but may also rely on Domain Name System (DNS) redirection to point the clients to suitable surrogates.

Content Selection and Management

Regardless of the content outsourcing conceptual category, the issue of content replication amongst a set of servers is a widely researched area, which started in the early 90s [96], and encompasses areas other than CDNs [97, 98]. [90] is one of the reference studies in content replication in the context of CDNs. In the paper, four heuristics are compared: random, local greedy, global greedy and popularity based, and the conclusions suggest that greedy approaches provide the best performance. This conclusion is supported by additional studies, each with their own particular insights and approaches [99, 100]. Apart from the random approach, most algorithms build rankings based on some metric such as content popularity, delivery latency [101], and surrogate load, to name a few, and then store the content according to these rankings.

The reason for not using the usually better greedy approach for content management lies in its high implementation complexity when compared to other simpler approaches that may rely on already existing server performance counters and logs, such as content popularity based on the number of requests for a particular content in a given period, server load, and so on.

Content placement is not an isolated issue, and should be seen as a component of the overall system, that ought to be optimized as a whole. In [102, 103] joint optimization of surrogate placement, content placement, and request routing are proposed.

The matter of what content to place on the CDN is not a trivial one, as several limitations usually inhibit the complete "CDNization" of HTTP traffic. The two main limitation are: dynamic and user-specific content is not easily cached and replicated; there are limited resources on edge/replica servers that must be properly managed in order to take the most out of the CDN. For example, we cannot just cache all the Internet's videos on a single replica server.

The issue of dynamic content is usually resolved by breaking coarse contents into smaller pieces where some of them are indeed dynamic but others are not. A common example is the header and footer of most websites which are the same regardless of what user is logged in. The "delta" in information between different users is most of the times minimal, and several content selection techniques rely on this fact to optimize otherwise seemingly dynamic websites [68, 104, 69]. Although conceptually simple, this approach

of breaking content into smaller cacheable parts is incredibly complex to perform in a standard/universal manner, as it depends on the content characteristics, and relies on processing-overhead vs. network bandwidth vs. storage trade-offs.

Another very important issue is that of caching techniques, i.e. given limited resources and a set of cacheable assets, decide which assets should be stored and which should be discarded at any given point in time, considering that adding or removing content from the CDN has a cost measured either in content access delay, server load, transit traffic, or even power consumption.

Caching

The performance of a content management system is highly dependent on the caching organization of the CDN, which encompasses the caching techniques in use, the update frequency, expiration policy, availability and reliability of content. Incorporating caching techniques in replica servers results in markable improvements in several key factors, including perceived latency, and reduced hit counts on the CDN backbone [105, 106], as shown in [106]. The authors suggest that the replica servers should implement caching policies in addition to their basic content-replication roles. A dynamic cache implementing "standard" caching algorithms such as Least Recently Used (LRU), Least Frequently Used (LFU) and SIZE [107], working along with the static cache represented by the static content that has been elected to be placed on a particular replica server provides improved performance, availability and resilience. Other alternatives relying on the same integration principle exist and may be found in [108, 109].

Caching in CDNs falls into two broad categories: *intra-cluster* or *inter-cluster*.

Several techniques are typically included in the *intra-clustering* category. In a *query-based* approach [110], in the event of a cache miss, the server broadcasts a request for the content to the other servers in the cluster and waits for the first *hit* response. This technique has the potential worst case scenario of having to wait for cache misses on all servers if the content is not found; hence, being prone to significant delays in the response, especially in the event of Denial of Service (DoS) attacks targeting non-existing contents, as the CDN server essentially acts as a request reflector.

An alternative method is the *digest-based* one [111], which solves the problem of message flooding in the event of cache misses. In order to avoid the flooding behavior, this method requires that each server maintains a digest of the content that is held by other servers in the cluster and provides for a more efficient cache-miss handling. Its main drawback is the challenge of keeping the digests up-to-date and synchronized between the multiple servers in a scalable, reliable, and low traffic overhead approach.

The *directory-based* approach [112] is another possible technique to use, that builds on the concept of the *digest* method, but instead of distributing the digests to every server in the cluster, places them on a centralized and shared directory to which the CDN servers access in the event of cache misses. Despite solving some issues of the *digest* approach, this centralized version has some issues of its own, namely: the potential bottleneck of querying the directory to locate the appropriate server from which to fetch the content: and issues of having a single failure point.

Another technique is the *hash-based* method, which derives the server hosting the content by hashing the incoming request [113, 66]. Because the content discovery information is indirectly contained in the request, this method solves some drawbacks associated with the previous techniques. The hashing function along with information of the servers on the cluster may be provided by a central entity, but does not rely on any highly dynamic data in order to function properly. The main issue with this method is that multiple requests for content that is hosted by a server other than the one storing the content will result in large amounts of redirects to the source server, with the potential of overloading the server in the event of some highly popular content.

To work around these issues, the *semi-hashing* scheme was proposed [114], and distinguishes itself from the *hashing-based* approach through the fact that a portion of its disk space is allocated to caching content that belongs to other source servers but was requested to this particular server. This behavior enables a reduction of the number of redirects for popular content so that events such as flash-crowds may impose a more balanced load on the cluster, instead of hitting a single source server.

In an *inter-cluster* scenario, where the clusters may be geographically distant, exhibit higher link delays or reduce bandwidth when compared to the *intra-cluster* situation, not all the previously discussed techniques represent viable approaches. For example, the hashing-based method will cause redirects to clusters which are probably on different geographic locations. The digest and directory schemes are also not appropriate as they require large content digests and directories which must be maintained; the common approach for inter-cluster scenarios is to use query-based methods [115].

Caching is by definition temporary and changing with time. Hence, mechanisms must be employed to ensure that the data being cached is relevant and not outdated.

One of the simplest update method is to use periodic updates, or expiration timers, where a timeout is defined specifying the time duration in which the content is still considered valid. Upon expiration the cache entry shall be removed or refreshed. This approach guarantees that cache entries are being updated regularly, but may suffer from excessive traffic due to cache refreshes in the event that the expiration timers do not match the "real" expiration of the entry, i.e. the point in time where the cache entry has become outdated, considering the original content source.

To work around the inefficient polling of the original source for refreshes, three techniques may be employed: *update propagation*; *on demand updates*; and *invalidations*.

An *update propagation* may be triggered on the event of an update of the content, and assumes that the caches will be notified that the content has changed; therefore, forcing a refresh of that particular content.

The *on demand update* method checks for content changes at each request but only updates the cache if the content changes, to guarantee that the client is always served with the latest version. This scheme requires management traffic to be sent back and forth from the caching server to the source server; therefore, it may be sub-optimal and highly inefficient if the content being cached is very small (eg: smaller than the traffic amount exchanged between the caching server and the source server).

Finally, the *invalidation* technique relies on the content server to send an invalidation

message to the caching servers which will immediately remove the content from their caches and repopulate it later on, if any request is made for the content.

Although these are the commonly employed techniques for cache management, combinations of the previously described approaches may be employed, sometimes associated with heuristics, to give the content providers full control over the content management policies on the CDN.

2.3.3 Request Routing System

A request routing system is a critical part of a fully featured, high performance, CDN. It is responsible for routing a client's request to the best replica server able to serve the request, and comprises a collection of servers, and/or network elements, supporting request-routing.

The best, or "closest" server suitable to accommodate the request is determined through a set of algorithms specifically designed for the purpose. In the context of request routing, the best server is not necessarily the closest one in terms physical distance [116]. The selection process must consider network proximity (how many hops away), the client perceived latency, bottleneck links, and server load for instance.

The request routing system does not follow a "one size fits all approach", as it is heavily dependent on the approaches taken on the content delivery and management system; thus, the algorithms and mechanisms used by the request routing system must take the CDN structure into consideration.

As an example, a CDN that uses full content replication (i.e. a given resource is fully replicated on the participating replicas) may simply perform a client redirect to the replica server hosting the complete content; whereas a CDN where partial content replication exists, and a resource is spread out through different servers, the request routing mechanism must act accordingly and route different parts of the resource to different replica servers.

Broadly speaking, the request routing system has two main components: the request routing *algorithm*, and the request routing *mechanism* [117].

The *algorithm* is triggered upon the client request, and determines how a request routing server should process the request and select a given edge server to accommodate it. In contrast, the request routing *mechanism* is the process through which the request routing server informs the client of its decision, right after performing the request routing algorithm.

Because every incoming request must go through this redirection system, its performance is critical and the implementation must be very efficient.

Figure 2.9 illustrates the steps involved in a possible request routing process (other algorithms / mechanisms may be used).

Request Routing Algorithms

The request routing algorithms fall into two broad categories: adaptive and non-adaptive.

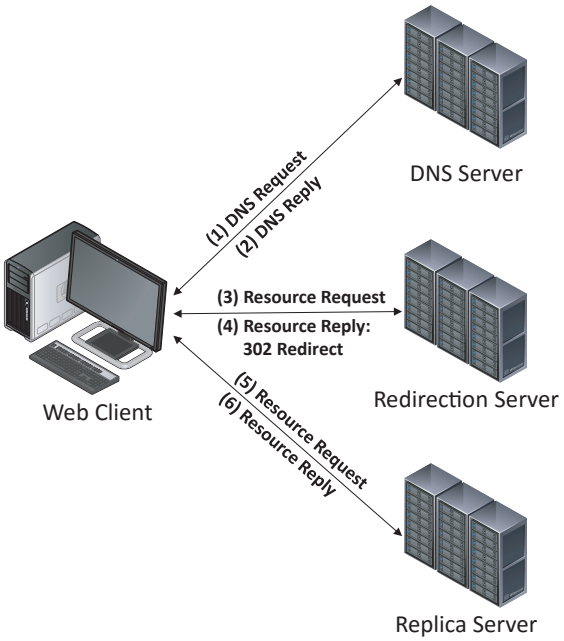


Figure 2.9: HTTP Redirect-Based Request Routing.

Adaptive algorithms, as the name suggest, are able to adapt to changing conditions of the CDN in the process of selecting a replica server. In this case, the algorithm uses as inputs metrics such as replica server load, distance to client issuing the request, and network conditions (considering congestion for e.g.) to name a few. Business logic may also be a part of the algorithm decision process, given that a specific replica server may be more expensive to use than others. Umbrella CDN [118] is an example of a CDN request router able to accommodate these kinds of business decisions.

The alternative non-adaptive algorithms follow strict pre-configured rules, or simple heuristics for selecting a replica server, and are much easier to implement.

Adaptive algorithms are more complex because they are required to change their behavior to cope with different situations, and exhibit high robustness in the event of flash crowds for example [119]. On the other hand, non-adaptive algorithms are efficient when the assumptions made by the heuristics hold true.

To further clarify the difference between the two types of algorithms, take as an example the round-robin algorithm, commonly used to distribute load among a set of replica servers. The algorithm tries to load balance the requests between the servers [120], under the assumption that they all have similar processing capabilities, and that any server may serve any client request. These are fair assumptions for request routing in the scope of a given cluster, where the replica servers are co-located and usually identical, but it is generally a poor assumption in wide area distributed systems, composed of several clusters, distant from each other. Each cluster will exhibit its own characteristics.

Considering only the distance aspect, clients that are redirected to farther replica

servers will likely suffer from poor perceived performance, given the increased latency. Additionally, the load balancing aspect may not be fully accomplished, given that blindly routing requests without considering their computational costs may overburden some servers when compared to others that have received the same number of request, but have lower computational cost.

Another example of non-adaptive routing that takes into consideration the distance of the client to the replica servers, and the predicted server load, is presented in [121]. The server load is predicted based on the number of served requests. These two parameters are taken as the two most influencing aspects conditioning the performance of the system. Despite being a marked improved from the round-robin scheme, the exhibited performance may still be poor.

Alternative implementations rely on hashing mechanisms, similar to the ones of DHTs, where the hash of the incoming request is determined and based on the result, the server that is closer to the computed hash in the hash ID space is selected [113].

With respect to adaptive algorithms, a vast number of proposals exist [122, 123, 124] that converge in the sense that a distance metric is computed between the client and the potential replica servers, in order to determine the replica server with the lowest (or highest, depending on the approach) metric to the client.

This metric typically takes into account aspects such as the location of the client and of the replica servers, network congestion estimates between the client and the replicas, the replicas' current load and their capacity. Given the additional information used by adaptive algorithms, they perform better that non-adaptive ones.

Besides the added complexity in devising and implementing the adaptive algorithms, there are other drawbacks. A significant one is that frequent probing of the network is required in order to gather statistics, generating network overhead. The approach taken by Akamai to gather these statistics is to establish virtual clients that request content from the CDN. Although passive approaches to network state measurement are possible, such as the ones used in Globule [122], they have been shown to be inaccurate [125].

The described algorithms are all server-side, given that usually CDN providers require full control on the selection process, nevertheless, client-side request routing is also possible and has been studied. In [126] a survey is presented regarding client-side request routing algorithms. This approach relinquishes the replica server choice to the client, which is given a list of the replicas that can provide the requested content. Upon receiving the list, the client may then decide the resource acquisition process, which may either use a single replica server, or multiple replica servers.

The multiple-replica server approach may require the content to be divided in blocks, so that parallelism is achieved in the download and reassembly process. A comparison between these two approaches is performed in [127], which concludes that on most scenarios the single replica server selection approach is preferable, with the exception of cases where the content to be downloaded is very large.

Request Routing Mechanisms

Having analyzed the question of *where to* route the client requests, the question of *how to* perform the actual routing remains. This section will address this issue and explore the predominant approaches.

The most popular approach relies on DNS redirection [128], which is used on Akamai for instance [129]. In this scheme, the CDN providers' controlled DNSs are responsible for running the request routing algorithm and mapping the symbolic request name to the appropriate replica IP address. This efficient approach, that avoids HTTP redirects and is transparent to clients, has been evaluated in several studies which culminated in a very detailed RFC 3568 [130].

Despite its advantages, some drawbacks exist regarding this approach. Firstly, the potential for increased network latency due to an increase of the DNS lookup times exists. Secondly, there is the issue of DNS resolution caching at the client and ISP level that may lead to sub-optimal load distribution between replicas. Thirdly, in the event of replica server failures, the DNS cache may lead to requests not being fulfilled in spite of other replica servers/clusters being available, especially if the DNS resolution process only returns a single A-Record.

Because of these downsides in the DNS resolution process, DNS based request routing is usually performed in a highly "coarse" manner, i.e. to help narrow down the subset of replica servers that will be used to process the requests, at the regional or country scale. After this initial redirection, another request redirection server will perform a local, more accurate redirection decision.

HTTP redirection represents another possibility for request routing. In this technique a server responds with a 301 (Moved Permanently), or 302 (Found) status code informing the client that the resource requested is available at a different location. This approach provides great request routing flexibility, and does not exhibit the downsides of DNS-based request redirection, although it does require and extra round trip for each redirected resource, which adds latency overhead to the request.

Another commonly used mechanism of request routing relies on Uniform Resource Locator (URL) rewriting to replace URLs contained in a main resource (such as a web page) by modified URLs targeting replica servers, i.e, the client fetches the initial web resource from the request routing server, and then the associated resources are requested directly to the replica servers/clusters. This approach is subdivided into pro-active URL rewriting, when the resources are static and preprocessed before being served, and dynamic URL rewriting which is performed in a per-request base. Naturally, the latter adds overhead to the process and may become a bottleneck if the rules/scripts are too costly to perform.

Other mechanisms include Global Server Load Balancing (GSLB) [131], anycasting [132], and CDN peering [133].

2.3.4 Performance Measurement

As with any performance-critical service, measuring the performance of a CDN is of utmost importance, given that it heavily impacts key factors such as: user experience (QoE), traffic billing due to inefficient use of bandwidth, and required number of servers, which affect CAPEX and OPEX.

A set of five key metrics are usually established in order to assess the performance of a CDN [56, 134, 135]. As with any system relying on caching, the first key metric is the cache hit ratio, which reflects the ratio of cache hits vs cache misses. The higher this metric, the better the caching algorithm and more clients are served directly from cache without any need for the acquiring the content from origin servers.

Next, the bandwidth used by the CDN replicas and origin servers in the content replication procedures. This metric must be under control both for performance reasons and for cost reduction purposes.

The overall latency that is perceived by the end-user is another crucial factor, as it directly impacts the users' QoE and ultimately the usability of the CDN solution.

Server utilization provides another key insight on the performance of the CDN (Central Processing Unit (CPU) load, Random Access Memory (RAM) usage, Hard Disk Drive (HDD) IOPS, ...), and influences the previously described factors

Lastly, the reliability of the CDN should be evaluated, through packet-loss measurements for example, to ensure that the servers are always available to their clients.

Several CDNs rely on these metrics to perform self-optimization procedures, such as avoiding redirects to replica servers that are overloaded, or selecting the replica servers that minimize the perceived client performance.

These key metrics may be extracted through several methods, which are generally classified as being either internal measurements or external measurements.

The internal measurements class relies on the CDN servers to gather data and make it available for performance monitoring tools. Examples of internal measurements include performance counters, server logs, internal probes, and Simple Network Management Protocol (SNMP).

External measurements usually complement internal measurements and are performed by third-party entities [136, 137]. A complete performance evaluation requires both internal and external measurements.

In addition to measurements, evaluation through simulation is also a viable approach for testing and estimating the impact of changes on the CDN [138].

2.3.5 CDNs and Multimedia Streaming

CDNs have been developed with the general purpose of serving content to Internet users, and have succeed in providing reliable and scalable services. In spite of the obvious benefits of using CDNs to serve multimedia content in general, and Catch-up TV in particular, this content exhibits a dynamic demand behavior with characteristics that require specific features and tuning [139, 52].

CDNs for general web content are typically limited to on-demand progressive streaming, usually in the form of low-quality streams. These characteristics limit their usefulness for live streaming services, and for high-quality media streaming.

Take as an example the streaming of high-quality multimedia content. The complete asset is large (hundreds or thousands of MegaBytes (MB)) and may be cumbersome to manage: it occupies a significant amount of storage on the servers' HDDs, thus reducing the number of items that a given server may hold and potentially limiting the cache hit ratios; it has a significant impact of the traffic volume of the network due to its size; and may reduce the user QoE in the event of a cache miss, given that it will take a lot more time to populate the replica server with a large file than with a small one. Additionally, if the user does not require the complete content, as is usually the case [140, 141] when he or she skips portions of a video, a lot of resources are wasted.

Providing live streaming services is another big challenge when CDNs are being used, as these types of services require a well-controlled end-to-end delay, in order to still be considered "live" [142].

To work around these limitations several streaming technologies were developed that able to leverage key CDN characteristics to provide the best possible user experience for a given context, such as Microsoft Smooth Streaming [37], Apple HTTP Live Streaming (HLS) [143], or Moving Pictures Expert Group (MPEG)-DASH [22]. This is not a trivial task as it has an impact on several CDN design decisions, such as the ones described in 2.3.2. Examples of multimedia-tailored CDNs are presented in [144, 145].

2.3.6 Optimization, Management & Provisioning

The rapid evolution of CDNs is strongly tied to developments in the cloud computing area, which has grown tremendously in size, usage and complexity.

One of the key challenges in cloud computing and, consequently, in CDNs, is that of proper resource management and provisioning, which must ensure that the running services have suitable amounts of computational, storage and networking resources at their disposal, without excessive and costly over-provisioning.

While static optimizations are possible, by thoroughly analyzing past demand data, they are error prone and subject to human-error. A more interesting scenario with potentially higher efficiency gains is that of autonomic and dynamic CDN optimization, capable of providing better resource usage, lower costs, and power consumption.

Even though the issue at hand is focused on multimedia environments, specifically on the Catch-up TV service use-case, the more general issue of dynamic and autonomic cloud resource management has been widely explored. [146] provides an overview on the issues and direction of cloud resource orchestration, which stresses the difficulties associated with dealing with pervasive, highly dynamic and heterogeneous cloud computing resources requiring expert knowledge for deployment, maintenance, monitoring, and control tasks. The need for a resource orchestrator able to forecast and adapt to changes in applications behaviors is identified as a crucial component of the resources' management process.

The work in [147] identifies the need for dynamic network resource provisioning as essential to maintaining a high-QoE in entertainment systems. In this paper, the authors propose the inclusion of a management and control plane responsible for holding a resource prediction engine, combining long and short-term forecasts for resource utilization which is reused to decide the optimal delivery approach, such as using CDN nodes, or engaging in P2P distribution

[148] conducts a survey on forecasting and profiling models, which frames the relevance of the problem at hand and systematizes the key motivations behind these techniques, namely application management, resource management, and cost management. Autonomic resource management is well represented by the MAPE-K (Monitor, Analyze, Plan, Execute, Knowledge) autonomic loop [149], and its related *self-** challenges.

[150] approaches the issue of dynamic resource provisioning in data centers through a reinforcement learning system aiming to reduce job rejection, as its primary goal, while simultaneously minimizing the overall energy consumption, as a secondary and conflicting goal. The results show that the use of machine learning to intelligently manage jobs mostly eliminates job rejections while reducing the total energy consumption.

In addition to the dynamic and autonomic management challenges, the scientific and industrial community has identified content-awareness as an effective way of improving new and existing systems [151, 152, 153].

Content-awareness refers to the adaptation of data storage, processing and/or transmission methods according to characteristics of the content being delivered to end-users. This process is highly dependent on the delivery systems' ability to extract meaningful information from content that is suitable for context-specific optimizations. The application of content-awareness to CDNs has been explored to improve multiple aspects such as request-routing, network planning, load-balancing, node placement, and caching.

In [154], an all-encompassing approach is taken to simultaneously solve the problems of request routing, node placement, and content eviction. The authors abstract the CDN as a switch-scheduling problem and proposed 3 different algorithms inspired in the Max-Weight scheduling algorithm, where content popularity is inferred by analyzing the request queues. In this case, the content-awareness refers to the fact that each source is aware of every item held by the caches, which poses a distributed knowledge synchronization problem that is not easily solvable for large, heterogeneous, CDNs.

[155] proposes a multi-criteria optimization algorithm in a scenario where information comes from multiple sources, with the purpose of jointly optimizing the selection of the best delivery server and path. The issue of multi-criteria optimization is clearly presented, along with the definition of what is an *efficient* solution. Substantial gains are shown by applying the proposed decision criteria; however, the baseline comparison is performed using random server selection, which is not representative of commercial delivery solutions.

Mangili *et al.* [156] focus on the specific issue of content-aware network planning, with the purpose of modeling and studying the migration to future Information-Centric Networkss (ICNs). Using a Mixed Integer Linear Programming (MILP) model formulation, their findings suggest that the migration of a small set of agnostic nodes to content-

aware ones is enough to provide substantial traffic reduction benefits to operators and content-providers.

The authors of [157] propose a content-aware dynamic load-balancing algorithm capable of taking into account not only the servers' load, capabilities, queue lengths, and historical performance, but also content characteristics, regarding their computational and bandwidth impacts. The approach is shown to significantly outperform static load-balancing algorithms (e.g. weighted round-robin) in terms of content response delay.

A related work is presented in [158], where a content-aware traffic-engineering process, Content Aware Routing (CAR), is proposed that enables content request aggregation from multiple origin servers and is able to provide a balanced traffic matrix that minimizes link congestion.

2.3.7 Conclusion

Content Delivery Networks are a center-piece in Internet's modern content delivery architectures, with continuous evolution in the past two decades. In spite of their apparent simplicity, they represent a highly complex interconnection of servers, where issues comprising server placement, choice of caching algorithms, and request routing systems, to name a few, have a high impact on the performance of the overall solution.

In the context of OTT multimedia delivery systems, CDNs are required to provide the end-to-end scalability necessary to support millions of simultaneous users. This is not a trivial task, and requires a thorough understanding of every aspect of CDNs, which in turn must be custom-tuned to deliver multimedia content in an efficient and cost effective manner.

From the range of topics addressed in the literature review, the issues of caching, content management, multimedia support, and dynamic resource provisioning stand out as specific, and critical, aspects to improve towards next-generation OTT multimedia CDNs; thus, these topics will be directly addressed in this Thesis' research work.

Even though CDNs were initially developed in a content-agnostic mindset, their usage in the context of multimedia streaming technologies requires a deep understanding of the protocols in use, which are the focus of the next section.

2.4 Multimedia Streaming Technologies and Protocols

The goal of this section is to provide an overview of the technical characteristics, issues, and challenges of different streaming protocols, and how they evolved to the current state-of-the-art technology and standards.

Streaming may be defined as the process of transmitting data through a specific channel, or medium, to a receiving device able to consume the data while it is still being transferred, as opposed to the non-streaming scenario, also known as download-and-play, where all data must be transferred before being played back [159].

Multimedia streaming gained popularity with the appearance of RTP Streaming Protocol (RTSP) in 1998 [160]. At the time, one of the most critical performance aspect of video streaming was the bandwidth required to stream the content, given than even low resolution content (e.g. QCIF) when paired with low performance compression technologies was overwhelming to the slow Internet connections (typically dial-ups of 56Kbps) that most people had access to.

With the advent of high speed Internet connections such as DSL and 3G, and more sophisticated video compression techniques, video streaming began to be feasible and accessible to everyone, as the huge success of IPTV platforms and OTT services such as YouTube and Netflix demonstrate.

The relevance of video streaming services kept growing in the last decade, and TV streaming services became so widespread that most telecommunication operators have TV streaming offers of their own, both IPTV and OTT.

Generally, video streaming services use different types of media streaming protocols, which are categorized as *push-based* and *pull-based* protocols.

In push-based protocols, after the establishment of the client-server connection, the server maintains a session and streams the packets to the client until the connection is stopped or interrupted by the client. In pull-based streaming, the server remains idle, waiting for client requests. The most common protocol for pull-based streaming is HTTP [161].

The selection of the streaming protocol must also take into consideration the nature of the content itself, in particular, if the content is *live* or *on-demand*, and this characteristic creates boundaries that impact the technical implementation of the streaming protocols. As an example, on-demand content does not exhibit any particular time constraints or relevance; however, live streaming services should enforce maximum streaming delays, as they fall into the category of delay sensitive services.

The discussion in this section will first begin with traditional forms of streaming, i.e. RTSP, which has been around from the late 1990's [160], to proceed to the currently commonly used progressive streaming, and will end with an analysis of adaptive streaming protocols, with special emphasis on the HTTP-based variant of adaptive streaming protocols given its booming popularity and potential.

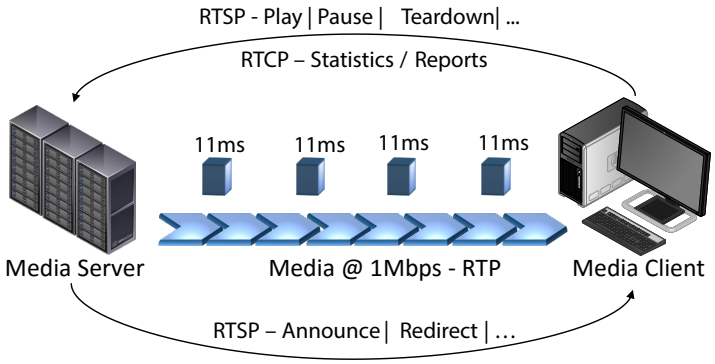


Figure 2.10: Traditional Streaming Using RTP Streaming Protocol.

2.4.1 Traditional Streaming

Multimedia streaming over the Internet grew in popularity with the rise of RTSP [160], used in conjunction with Real-time Transport Protocol (RTP) [162, 163] and RTP Control Protocol (RTCP) [163]. RTSP was designed to support and control the delivery of data for entertainment and communications systems with real-time properties, and supports "trick-modes" such as "play" and "pause".

Because it is a stateful protocol it is known as a "network remote control": the session state is maintained from the moment that a client connects to the streaming server until the connection is terminated, by issuing a "teardown" command.

When a session is established, the server starts streaming and sends packets to the client using an RTP data channel, either over User Datagram Protocol (UDP) or TCP.

To maintain a stable session, RTSP uses RTCP to collect QoS data such as bytes sent, jitter, packet losses and Round Trip Time (RTT), i.e. the streaming connection is bidirectional and requires server interaction in order to be able to support trick-modes and some basic forms of adaptation. This kind of server interaction adds to the overhead of the streaming session and limits its scalability [164].

Figure 2.10 depicts an example of RTSP streaming using a typical packet size of 1452 bytes for an average content bit rate of 1Mbps, along with some possible commands sent to/by the server from/to the client.

The data is sent in a paced manner and independently via RTP packets, which means that each packet contains information of about 11ms of video. The just-in-time delivery of RTP minimizes the bandwidth consumption, but provides a reduced margin for recovery in case of packet losses (a real possibility over UDP). To compensate for this fact, RTP ignores the lost packets and supports graceful degradation of the playback quality, avoiding playback stoppages in the case of non-critical packet losses [165].

The RTSP characteristics pose (at least) three challenges:

- Scalability: supporting millions of devices (for a service with the scale as YouTube, for instance) requires managing millions of sessions, and is infeasible in practice even if RTSP proxies are used;

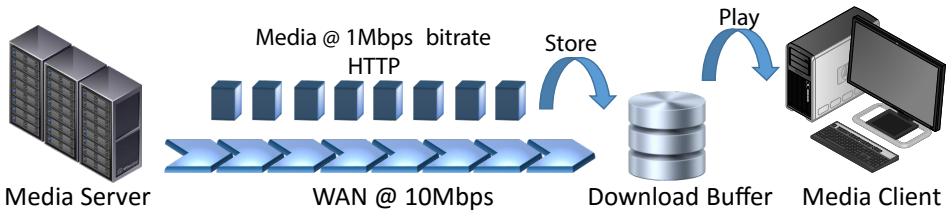


Figure 2.11: Progressive Download Example.

- Issues with managing the multiple port and protocol nature of RTSP: added complexity of connection establishment and QoS management;
- Firewalls and proxies, especially if UDP is being used as a transport protocol: there is no guarantee that there will an "open path" to the client, particularly when home gateways are involved.

These difficulties, along with advances in available network capacity and last-mile bandwidth, limit the usage of RTSP on modern services, although niche use-cases and markets still exist [166, 167].

2.4.2 Progressive Download

A pseudo-streaming method that gained traction in the past years is progressive download. This approach treats a streaming session just like regular/bulk data download. The term progressive has its root in the fact that as soon as the media player receives some data – i.e. the download is still in progress and the file has not been fully written to disk– the playback may begin. Figure 2.11 provides an overview of the progressive download process.

It is usually supported over HTTP and, in addition to direct browser support in HTML5, several vendors provide specific media players and technologies using this method, such as Adobe Flash, Microsoft Silverlight, and Windows Media Player. Many popular video streaming websites use this technology: e.g. YouTube, and Vimeo.

Because the playback is managed by the client, it is possible to use trick-modes, so as long that the requested part of the video has already been downloaded. Skipping and seeking to yet-to-download parts of the video is supported since HTTP 1.1.

The growth in popularity by this type of streaming was mostly due to two characteristics: easy firewall transversal, as HTTP is typically allowed in every firewall and proxy, and high scalability. Because file downloading through HTTP is stateless, it can easily use proxy servers and distributed caches or CDNs. This is essential when VoD is being delivered to millions of users, as most of the traffic is unicast.

The trade-off associated with this scalability is the loss of several features of RTSP:

- No support for live streaming;
- No adjustable streaming based on QoS metrics;
- No "instant playback" support, i.e. depending on the quality of the media and

the network's condition, it may take more time to download the content than to watch it;

- No graceful degradation. Missing packets will cause the playback to stop until the required data is downloaded;
- Waste of bandwidth: The average YouTube video length, as of 2007 [141] was of about 4m:15s; considering that 20% abandon a video after just 10 seconds, and 60% after 2minutes [140] there is a big potential for wasted bandwidth.

In spite of these disadvantages, progressive download technologies are widely used, and demonstrate the importance of scalability and firewall penetration.

2.4.3 Adaptive Streaming Technologies

In a scenario with unreliable or varying network conditions, as is the case of today's Internet, which is a collection of multiple networks all over the world, adaptation plays an important role in improving the perceived user QoE. The previously analyzed RTSP contemplates some adaptation possibilities via the feedback QoS metrics sent through RTCP, but the progressive download scenario does not.

Adaptation is a crucial feature of any streaming technology, as any degradation of the connection quality (in terms of bandwidth, latency or dropped packets) may cause dropped frames, freezes, and/or long buffering delays, and render the viewing experience unbearable, especially in the content being streamed is a live event [168].

There are several ways to provide adaptation, but the most relevant classes are:

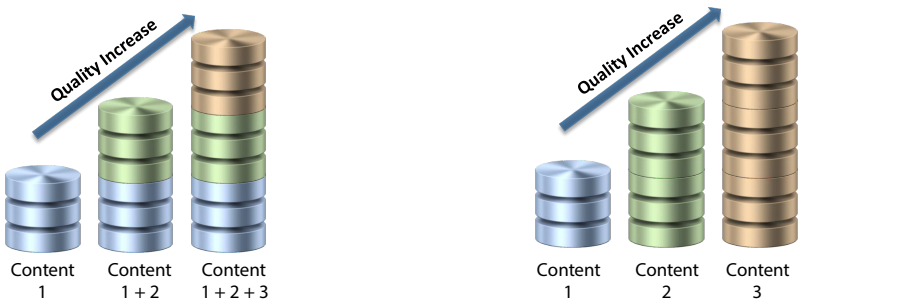
- Adaptation as a feature of the content encoding process, i.e. the content is encoded in a scalable manner with a baseline quality and with quality improvements if extra data is received (Figure 2.12(a)) - described ahead as the Source Video Coding class;
- Adaptation as a feature of the content distribution process if the content delivered is flexible in terms of quality/bit rate, i.e. if multiple representations exist for the same content but with different quality/bit rate levels (Figure 2.12(b)) - the dominant class is segmented HTTP-based delivery.

Hybrids exist that combine these adaptation methods [169]. The specifics of each class and related technologies will be depicted in the subsequent sections.

2.4.4 Source Video Coding

Source Video Coding represents a class of video encoding techniques that provide adaptability on the encoding block of the multimedia content creation and delivery pipeline (Figure 2.2). These techniques were developed with scalability and robustness in mind, so that the device receiving the content can still properly decode it in situations of varying network quality and availability, even if with a penalty in content quality.

There are currently 2 main scalable source video encoding techniques: Multiple Description Coding (MDC) and SVC.



(a) Scalability via Incremental Representations (b) Scalability via Multiple Representations

Figure 2.12: Adaptive Streaming Classes.

(a) Both descriptions. (b) Description 1. (c) Description 2.

Figure 2.13: Multiple Description Coding Example. Adapted from [2].

Multiple Description Coding (MDC)

The concept behind MDC is to generate multiple "descriptions" [170], so that each description contains enough information for the device to playback the content, whose quality improves with the number of descriptions received; hence, scaling in quality.

Media playback quality will be roughly proportional to the total available bandwidth of the content source. Figure 2.13 shows a sample of the impact of MDC in a case where 2 descriptions exist for a content. 2.13 *a)* shows the image restored from both descriptions, while *b)* and *c)* demonstrate the result of the individual decoding of each description.

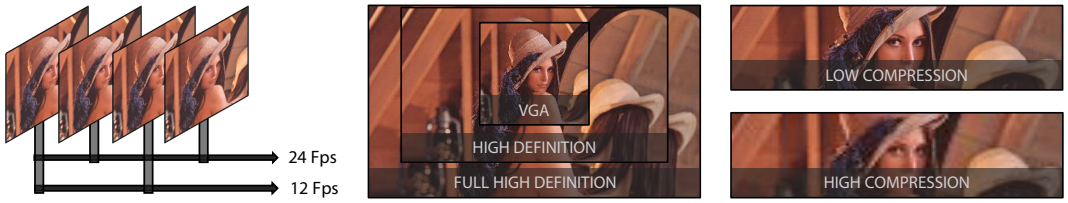
This methodology provides great flexibility in terms of the content source, as multi-homed or P2P networks, for example, are easily supported. In addition, given its resilient nature, the usage of best-effort network connections is a possibility that does not interrupt the playback unless every description is affected.

Several information-theory approaches to MDC exist, and vary on how they use the spatial and temporal information as well as pixel and frequency domain [170, 171, 172, 173]. Nonetheless, this approach presents some drawbacks. First, there is loss in compression efficiency due to redundancy, but the second and foremost disadvantage is that no standard has been established, which hampers its commercial deployment.

Scalable Video Coding (SVC)

SVC is a mature encoding scheme with an industry standard that has been finalized in 2007 (Annex G of the H.264/Advanced Video Coding (AVC) [174]). The standard promotes a high quality encoding of the original source in a scalable manner and with high encoding efficiency. Scalable, in the context of SVC means that subsets of the original stream may be removed, and the resulting sub-streams can still be decoded by receiving devices, albeit with an impact on the media's Frames per second (Fps), resolution and/or image quality. It supports format, bit-rate, and power adaptation, along with graceful degradation in lossy transmission environments.

Figure 2.14 shows 3 examples on how scalability might be achieved. The first example – Figure 2.14(a) – demonstrates how temporal scalability might be achieved by adjusting the frame rate; next, on 2.14(b) spatial scalability is attained by varying the resolution of the frames; and lastly, on 2.14(c), scalability in fidelity is achieved by adjusting quality parameters such as Peak Signal-to-Noise Ratio (PSNR) or compression ratios.



(a) Temporal Scalability (b) Spatial Scalability (c) Fidelity Scalability

Figure 2.14: Scalability modes of Scalable Video Coding (SVC).

These scalable characteristics make SVC a viable option for targeting different types of devices with the same source content. Mobile phones can use low-resolution / low-fps sub-streams [175] while desktop computers, for instance, may choose to use the complete original stream for the best resolution, frame-rates and image quality. Also, given this scalable nature, the receiving device may adapt on-the-fly to varying transmission conditions [176].

The disadvantage is that while it is based on the H.264/AVC standard, an SVC specific decoder is needed to take advantage of the scalability features, in spite of requiring only a small complexity increase on the decoder for proper support.

Conclusion

Techniques relying on scalable and complementary information have the potential to guarantee uninterrupted playback and provide a good user QoE in face of changing network conditions. MDC and SVC have different pluses and minuses, but SVC is clearly a technologically advanced technique given its small overhead (∼10% [176]) when compared to an H.264/AVC stream without the scalability extensions; in addition, it is a standardized technology.

2.4.5 Adaptive Segmented HTTP-based delivery

A more recent approach – when compared with source video coding – for delivering content in a scalable manner is to use segmented HTTP-based delivery. The idea behind this method is to encode the original content into streams of different quality, Fps, and/or resolution and then fragment those streams into segments - or "chunks" - , usually 2 to 10 seconds long, that can be individually downloaded and decoded (Figure 2.15).

Segmented HTTP delivery is the natural evolution of progressive download streaming: it provides the same benefits of progressive download without the drawbacks of not supporting adaptation or live streaming, while adding some new features of its own.

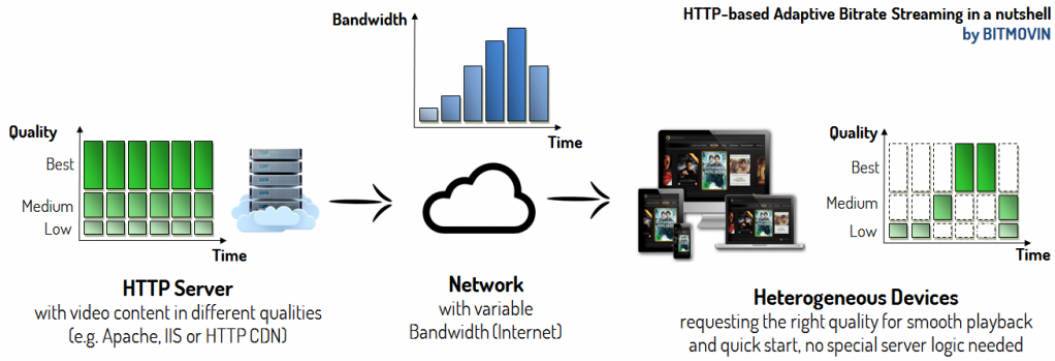


Figure 2.15: Segmented HTTP Adaptive Streaming [3].

By using this approach, the adaptation intelligence is relinquished to the client device, which has to monitor the connection quality and decide which chunk and stream to use at a given point in time. This is a big advantage as many issues generally exist in access networks, and the client is in the best position to assess the network quality [177, 178]. Moreover, this inversion-of-control approach enables the client to make decisions not only based on the network quality but also on its own computational resources, which may be limited, for instance in the case of mobile, old or low power devices.

The chunks of each stream are numbered and are time-synchronized with matching chunks in alternative streams; therefore, the client can mix and match chunks from different streams, and the only impact will be on the quality, Fps, or resolution of the media. The client has full control over the downloaded data and may buffer as much, or as less, data as it desires.

This adaptation method has a big advantage over source video encoding methods: because the delivery is independent of the encoding technique, as long as the encoders are able to produce atomic and individual chunks, there is a complete abstraction between the delivery method and the content itself. This decoupling makes the delivery method agnostic to the content and reusable for different encoding schemes. In fact, source video encoded content could also run on segmented HTTP-based delivery [169].

The popularity of this delivery approach grew as multiple vendors implemented their own version of segmented HTTP-based delivery technologies with slightly different char-

acteristics but, in the end, very similar between each other:

- Microsoft specified Smooth Streaming;
- Apple developed HLS;
- Adobe created HTTP Dynamic Streaming (HDS);
- MPEG standardized DASH in April 2012 [22].

Quoting Akamai's Will Law [179]: *"(the technologies) are 80% the same, yet 100% incompatible. To view HLS, you must have a player for that format. For HDS, another player and for SmoothHD, a third. This fractured delivery space forces encoders, delivery networks and client players to spread their development efforts across all these formats, forgoing optimizations that could be achieved by converging around a single format"*.

The ensuing sections will discuss HTTP delivery, and detail the specifics of each protocol so that a proper comparison can be made between the competing technologies.

Why HTTP Delivery?

HTTP delivery is at the core of each one of these adaptable technologies, and played an important role in the success they had.

Initial proposals to multimedia streaming had as its main challenges the networks' capacity and delays involved, and lead to the development of RTSP [160], a low-overhead streaming protocol with session/state-management features embedded (Figure 2.10). As the Internet developed, network capacity grew, HTTP became a commodity, and the big challenges in multimedia delivery shifted from the network to the servers' capacity: having servers managing separate streaming sessions for each client is not scalable and makes large multimedia content distribution deployments resource-intensive.

Considering that the Internet was essentially built around HTTP, it has become extremely optimized for this particular method of delivery, where large segments of data are being exchanged. The value of delivering small packets *per se*, such as TCP packets has diminished, hence the widespread use of progressive download technologies and CDNs to help deal with content locality and reduce the long-haul traffic in the Internet.

A multimedia delivery method using HTTP is then inherently taking advantage of the following facts:

- Most firewalls are already configured to permit HTTP traffic, i.e. TCP port 80, whereas in the case of other streaming protocols this might not be the case – easy firewall and NAT transversal;
- HTTP is known as being a stateless protocol; thus, the streaming session can be managed by the client instead of the server, relieving precious server-side computational resources. Each segment requested will require an individual, short-lived session;
- Reliability and deployment simplicity: HTTP and TCP are widely tested and supported.

Microsoft Smooth Streaming

Smooth Streaming [37] is Microsoft's take on adaptive segment-based HTTP Streaming. It builds on the concept of fragmented MPEG-4 standard [180], supports H.264/VC-1 as video codecs, and Windows Media Audio (WMA)/Advanced Audio Coding (AAC) as audio codecs.

It is essentially a proprietary solution, despite Microsoft's efforts to standardize it through Protected Interoperable File Format (PIFF) [181], and its active involvement in 3rd Generation Partnership Project (3GPP), MPEG and Digital Entertainment Content Ecosystem (DECE).

The technology has three main components:

- The encoder (usually Microsoft Expression Encoder, though other vendors provide compatible solutions, such as Envivio);
- A Microsoft's IIS Media Services extension which provides the streaming services to the clients' (third party companies also provide compatible streaming services, such as Wowza);
- The Smooth Streaming Client: Microsoft provides client implementations based on Silverlight that can be used on Microsoft platforms, as well as client porting Software Development Kits (SDKs), which have been used by third-parties to provide client implementations for popular devices such as the ones running Apple's iOS and Google's Android.

This general architecture is conceptually identical for every segmented HTTP-based streaming protocol. The encoder generates PIFF compliant content (which contains the media itself) as well as two additional files, called "Manifests", which are Extensible Markup Language (XML) – formatted files. There are two main types of manifests: client and server.

Server manifests provide a very high level perspective on the characteristics of the media file, and are used by the streaming service, usually IIS, as metadata that provides a macro description of the encoded content:

- Number of encoded streams (tracks);
- The track type: video, audio, or text;
- Location, as each track might be stored in a different file -– for on-demand content;
- Track content information – codec type, private data, bit rate, resolution, etc.

There are two variants of server manifests, depending on whether the content is live or on-demand, but the structure is essentially the same. A sample on-demand server manifest is shown in Listing 2.1.

As for the client manifest, it is must be downloaded by the client and processed in order to initiate the playback. Its data reveals the internal structure of the adaptive content, and provides complete information about the number of available tracks, their encoding, resolution, duration, and how they are fragmented (number of chunks and duration of each chunk – the default duration is 2 seconds per chunk).

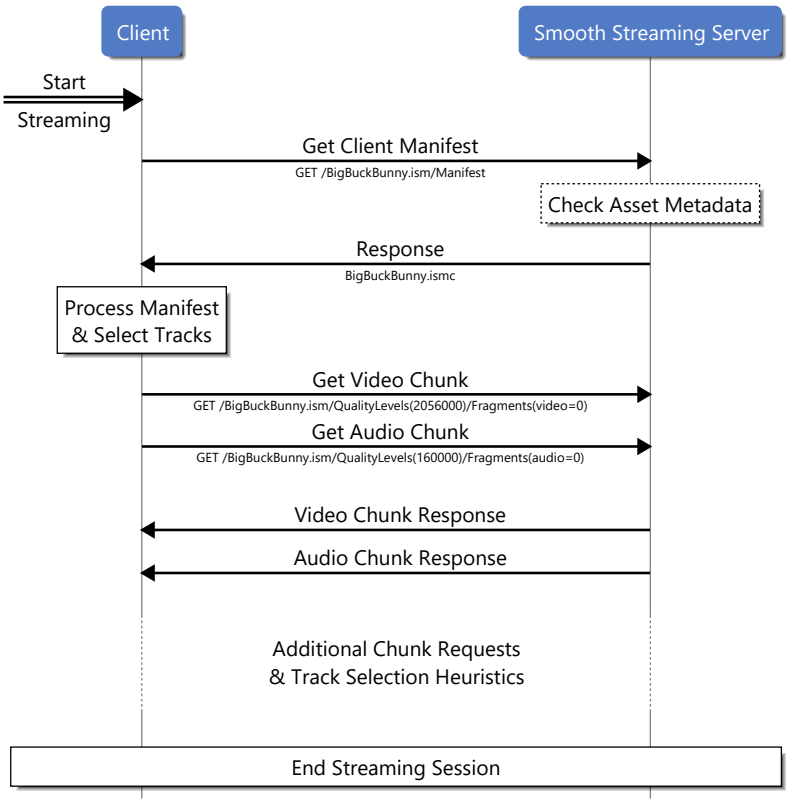


Figure 2.16: Simplified Smooth Streaming Session Diagram.

With this information at hand, a client may decide to first request the lowest quality chunks and then, after evaluating the response time of these initial chunks decide whether to scale up the quality of the requested chunks, or not.

A sample (trimmed) client manifest is shown in Listing 2.2, and a simplified playback flow is shown in Figure 2.16.

Apart from the adaptive streaming core advantages, Smooth Streaming has additional advantages with respect to extended metadata support, in the form of chapters, markers, subtitles, multiple audio tracks, Digital Video Recorder (DVR) buffer for live content and, most importantly, support for the widely industry-supported DRM technology Microsoft's PlayReady, which goes one step beyond the typical Initialization Vector (IV) / Advanced Encryption Standard (AES) - based DRM. In a digital world dominated by content providers, this is an essential feature.

```
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
  <head>
    <meta name="clientManifestRelativePath" content="BigBuckBunny.ismc" />
  </head>
  <body>
    <switch>
      <video src="BigBuckBunny_2962.ismv" systemBitrate="2962000">
        <param name="trackID" value="2" valuetype="data" />
        <param name="trackName" value="video" valuetype="data" />
      </video>
      <video src="BigBuckBunny_2056.ismv" systemBitrate="2056000">
        <param name="trackID" value="2" valuetype="data" />
        <param name="trackName" value="video" valuetype="data" />
      </video>
      <audio src="BigBuckBunny_2962.ismv" systemBitrate="160000">
        <param name="trackID" value="1" valuetype="data" />
        <param name="trackName" value="audio" valuetype="data" />
      </audio>
    </switch>
  </body>
</smil>
```

Listing 2.1: Sample Smooth Streaming Server Manifest.

```
<SmoothStreamingMedia MajorVersion="2" MinorVersion="1" Duration="5964583334">
<StreamIndex Type="video" Name="video" Chunks="299" QualityLevels="8" MaxWidth="1280" MaxHeight
    ="720" Url="...">
<QualityLevel Index="0" Bitrate="2962000" FourCC="H264" MaxWidth="1280" MaxHeight="720" />
<QualityLevel Index="1" Bitrate="2056000" FourCC="H264" MaxWidth="992" MaxHeight="560" />
<.../>
<c d="20000000" />
<.../>
</StreamIndex>
<StreamIndex Type="audio" Index="0" Name="audio" Chunks="299" QualityLevels="1" Url="...">
<QualityLevel FourCC="AACL" Bitrate="160000" SamplingRate="44100" Channels="2" BitsPerSample="
    16" PacketSize="4" AudioTag="255" />
<c d="20201360" />
<.../>
</StreamIndex>
</SmoothStreamingMedia>
```

Listing 2.2: Sample Smooth Streaming Client Manifest.

Apple HTTP Live Streaming (HLS)

HLS is an Internet Engineering Task Force (IETF) Draft [143] that shares many similarities with Microsoft's Smooth Streaming; however, some minor differences exist: first, it only supports H.264/AAC encoding, and requires encapsulation using MPEG-2 Transport Stream (MPEG-2 TS) instead of PIFF fragmented MPEG-4; second, the structure of media metadata is described through a hierarchical use of "m3u8" play-lists.

A top level "playlist-file" (Listing 2.3) describes the existing tracks – "media-segments" – (Listing 2.4) in terms of content type and bit rate, and also specifies the content encryption, if any. Each media-segment has the pertinent information regarding each media track, such as each segment duration (usually 10 seconds) and additional information in tags.

```
#EXTM3U
#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=200000
gear1/prog_index.m3u8
#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=311111
gear2/prog_index.m3u8
#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=484444
gear3/prog_index.m3u8
#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=737777
gear4/prog_index.m3u8
```

Listing 2.3: Sample Apple HLS M3U8 Top Level Playlist.

```
#EXTM3U
#EXT-X-TARGETDURATION:10
#EXT-X-MEDIA-SEQUENCE:0
#EXTINF:10, no desc
fileSequence0.ts
#EXTINF:10, no desc
fileSequence1.ts
#EXTINF:10, no desc
fileSequence2.ts
...
```

Listing 2.4: Sample Apple HLS M3U8 Track Playlist.

Technology-specifics aside, HLS is the only adaptive streaming protocol supported by default on Apple devices; hence, it is suitable for content delivery in these environments.

The fact that it uses an encapsulation different that Microsoft's Smooth Streaming is in most cases only a nuisance, given that the encoding method is H.264 / AAC which is also supported on Smooth Streaming. Microsoft even allows for real-time / on-the-fly repackaging of Smooth Streaming live streams into HLS on its IIS streaming server.

The main disadvantages of HLS have to do with lack of proper DRM support, as it only supports simple AES content encryption, the lack of additional experience enriching metadata, such as markers, chapters and so on, and a generalized lack of compatible players, although a few third party ones exist for Linux / Windows environments [182].

Adobe HTTP Dynamic Streaming (HDS)

Adobe developed its own container for segmented media, F4V, which holds H.264 / AAC encoded content in chunks, based on the ISO/IEC 14496-12 MPEG-4 Part 12 standard [180]. The full specification was defined by Adobe and is not standardized, although a public document exists describing the media files' internal structure [183].

This streaming protocol also relies on: manifests describing the existing tracks - the F4M [184]; index files, to identify the position of a segment within a stream – F4X's; and segments - F4F's.

The only officially supported media player is Adobe's own Flash Player or players based on Adobe Integrated Runtime (AIR) technology, which limits the number of supported devices, albeit media servers like Wowza [185] are able to take advantage of the fact that the inner content is encoded in H.264/AAC to repackage it on the fly and support Smooth Streaming or Apple HLS from content based on HDS, though usually at the expense of DRM.

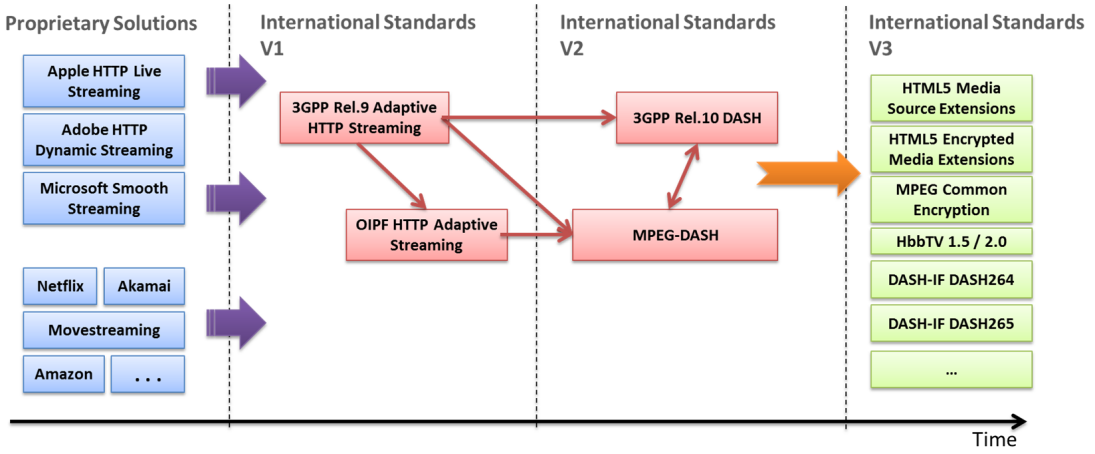


Figure 2.17: Evolution of Adaptive Streaming Protocols and Standards [3].

MPEG Dynamic Adaptive Streaming over HTTP (DASH)

Having covered the main competing technologies in the Adaptive Streaming segment-based HTTP delivery field, one last technology must be mentioned: MPEG-DASH.

This MPEG standard (ISO/IEC 23009-1:2012 [22]) was created to deal with issues associated with vendor-centric solutions, such as the ones enumerated so far. Creating an industry standard, without requiring vendor-specific ecosystems, enables the content distributors to focus more on the content itself and less on technological peculiarities and interoperability issues that rise on fragmented ecosystems, which is paramount, especially if we take into consideration that the vast majority of traffic in the Internet is video [15]: the video encoding, storage, distribution, and playback process must be streamlined and universally supported.

The companies that initially specified their own HTTP adaptive streaming protocols also realized these requirements, as the list of partners that have contributed to the MPEG-DASH specification includes, but is not limited to, Microsoft, Apple, and Adobe.

These reasons also led 3GPP to add support for DASH on its Release 10 specification [186], with further improvements on Release 11 [187]. In recent years, support for DASH has been extended to HTML5, under Media Source Extensions (MSE), and Encrypted Media Extensions (EME). Figure 2.17 illustrates the evolution of adaptive streaming formats and standards with time.

DASH uses Media Presentation Descriptions (MPDs) to describe time *Periods* (defined by a start time and duration) whose purpose is to facilitate the insertion of different media sequentially, so that scenarios like ad-insertion are possible.

Each time period holds information regarding *Adaptation Sets*, which in turn contain segment information. The adaptation sets contain the encoded alternatives (*Representations*) of a media component, while the segments represent the actual media data. The adaptation sets may contain any media data in its segments, as the technology is video/audio codec agnostic. In addition to the two types of recommended containers

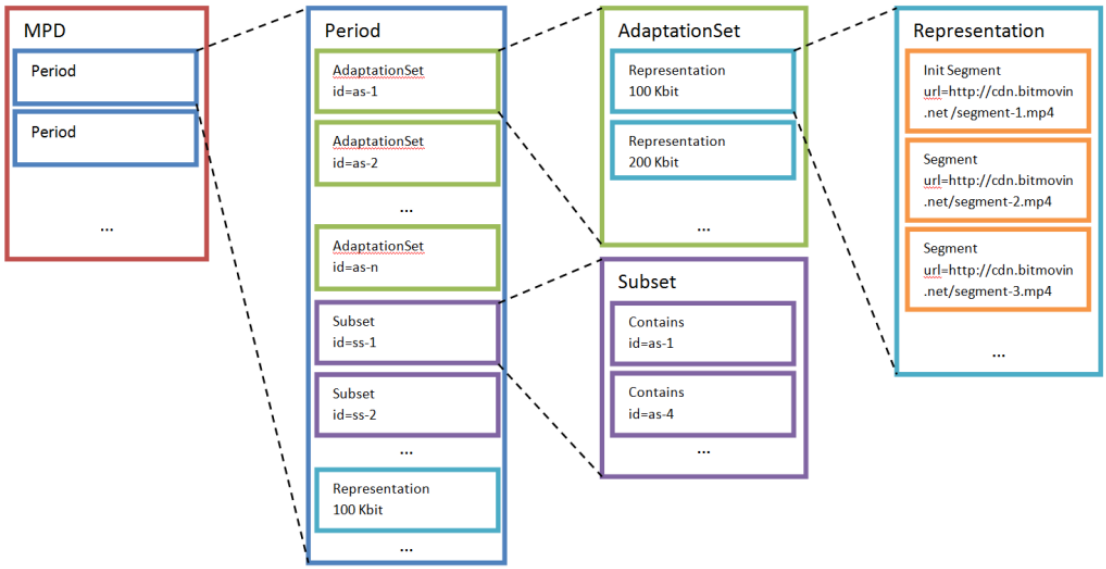


Figure 2.18: MPEG-DASH Media Presentation Description (MPD) Hierarchy [3].

(MPEG-4 and MPEG-2 TS), new formats relying on H.265/HEVC are also supported. Figure 2.18 provides an overview of the hierarchical layers of a DASH MPD.

MPEG-DASH provides an extensive list of features that draws the best from the preceding technologies while adding some new features of its own, such as:

- Seamless advertisement insertion for both live and on-demand content;
- Stream switch, for multi-camera view, multiple audio languages, 3D, and, naturally, multiple bitrates;
- Fragmented MPD, for composing MPDs using multiple sources;
- Alternate URLs, that allow a client to choose the best suiting content source, which is useful in the context of geolocation optimization, CDNs, or simply load-balancing;
- Support for SVC and MDC;
- Versatile set of descriptors, that can include content metadata, such as content rating, accessibility features and audio channel configuration;
- Support for quality metrics, so that the client may report predefined key metrics to the server;
- Segments of varying duration;
- Clock drift-control for live streaming;
- Flexible DRM support (different DRMs in the same MPD, pay-per-quality, ...).

As far as DRM support is concerned, MPEG-DASH is designed to support multiple DRM and Common Encryption (ISO/IEC 23001-7 [188]). Multiple DRM is supported

given that each adaptation set may use a DRM scheme independently of other adaptation sets in the same MPD, and as long as the client devices supports one of the specified DRM technologies, it will be able to decode the content.

As far as widespread adoption and research focus is concerned, the fact that this is an open standard, already with significant research initiatives, and that the DASH Industry Forum (DASH-IF) has as its members most of the world's top technology companies, the potential for MPEG-DASH to become the de facto HTTP based segmented streaming technology is significant.

Conclusion

The 4 main competing technologies in the segmented HTTP-based delivery streaming method were analyzed and their features, overall advantages, and disadvantages were explored. Given that they all share the same underlying vision, their generic top-level working mechanism is essentially the same, though with slight but significant differences that make some more compelling to use than others.

To summarize, Microsoft's Smooth Streaming is probably the most widespread technology in use with proper support for the commonly used PlayReady DRM, and with media player implementations for virtually all platforms.

Apple's HLS appears to be inferior to Microsoft's Smooth Streaming in some aspects, namely: DRM support, extensibility metadata, and client support; though, its playlist-like media description provides some flexibility with regards to mix-and-matching different content sources for inserting advertisement, or creating on-demand media clips from different sources.

Adobe's HDS is not as commonly used as other competing technologies, nor does it provide any differentiating factor that could make it a compelling proposition.

Lastly, MPEG-DASH, despite suffering for a lack of widespread adoption, partially because it is a recently standardized technology, is the proposition that appears to have the most potential value. Regarding client support, 3GPP has already included support for DASH in its latest releases, HTML5 already supports it, and efforts are being made to create clients for the different platforms. A big plus for MPEG-DASH is its great flexibility in terms of media composition, supported codecs and DRM. With the right push from the major companies, it is expected to become the *de facto* standard of HTTP adaptive streaming, with benefits to all: content providers, service providers, and the consumers.

2.4.6 Network and Client Adaptation Challenges

The main point of adaptive streaming as implemented by segmented HTTP-based streaming and source video coding technologies is to increase the degrees of freedom over which a client media player may act in order to adjust the media playback to suit its contextual conditions and capabilities.

In order to do so effectively, it is necessary to first know the environment, and then act according to it, and according to the users' expectations when using the application

[178]. The following bullet points provide a list of variation vectors:

- *Client*
  - Supported codecs and plugins;
  - Screen size & resolution;
  - Available computational resources.
- *Network*
  - Access type - wired or wireless;
  - Access technology - e.g.: Wi-Fi or 3G;
  - Quality metrics - Delay, Jitter, Packet Loss, ...

The video player is expected to use these inputs to maximize the overall QoE. In practice, QoE optimization is reflected on concrete technical actions, such as:

- Maximization of bandwidth usage, subject to network conditions - to provide the best possible viewing quality / experience ;
- Ensuring that video playback stutter or breaks do not occur, due to buffer underruns for instance;
- Avoiding frequent/fast oscillations in the stream quality, to maintain the perceived stream quality [189, 190].

An intelligent leveraging of the client attributes is important not only to provide a great QoE to the user watching the content, but also to avoid wasting precious network resources whenever possible. As an example, if a user is watching a video stream without any constraints, i.e. with the ability to watch the content at the maximum available bit rate, it does necessarily mean that he should, especially if he is not taking advantage of the full content resolution, such as in cases where he's not watching the video in full screen, or when he's running low on battery. These use-cases should be accounted for in an intelligent adaptive streaming.

As for varying network conditions, it is also important that the adaptive streaming process takes into consideration the particularities of each access technology and acts according to it. Take the following case as a common example: if the user is running on a 3G or 4G data connection and significant variations are detected regarding the network delay, jitter or even connection drops, then it is likely that the user is mobile and the client player should take proactive measures to opportunistically fill its buffers (maybe increase the buffer size?) when network connections improve so that on the event of connection drops the buffer will have enough data to compensate. In a wired environment, with stable QoS metrics the player might instead decide to be more aggressive on the quality adjustment algorithm in detriment of having a long and stable playback buffer of lower quality segments. These are the kind of decisions that a next-generation adaptive streaming player is expected to perform, to endow its users with the best trade-off between playback quality and playback stability.

With respect to the application of HTTP adaptive streaming technologies on 3G/4G networks, several studies have been conducted to assess the performance of available

commercial solutions [191, 192, 193, 194].

In a study conducted in Oslo, Norway, the authors of [194] perform an extensive evaluation on commercial adaptive HTTP streaming solutions in 3G network under different mobility cases: bus, ferry, metro, and tram. The conclusions of the study support that while the underlying technologies are essentially the same, the client player's implementations vary greatly on behavior: some privilege stable quality (e.g: Apple), Adobe's implementation provides the best possible video quality at a given point in time, while Microsoft provides a compromise between the two. According to the results, a clear winner does not exist as each implementation has its own strengths and weaknesses which vary with the network's conditions.

Other studies regarding the general performance of segmented HTTP-based streaming have been performed [191], with a stronger focus on the underlying delivery protocol (TCP), its behavior and how the player uses the TCP connections to explore the available bandwidth to the fullest and tries to avoid slow-start situations. Apart from the performance conclusions of each player, this study and its related work [195] indicate that TCP's behavior and how it's handled can make or break a good video player, in spite of the design of its adaptation algorithm. According to the authors: *"TCP provides good streaming performance when the achievable TCP throughput is roughly twice the media bit rate"* [191]. These results underline the need for advanced adaptation algorithms for client player use that are aware of its underlying technology limitations.

2.4.7 Live Streaming over HTTP

Streaming live content has been out of practical reach for technologies that did not rely on RTSP in some manner, and is one of the major shortcomings of progressive download. This fact changed with the advent of segment based HTTP streaming, though some limitations vs. RTSP exist, namely an increase in the stream delay, or latency.

To better understand the fine details and implications of live streaming over HTTP adaptive streaming, [142] presents an end-to-end overview of the live streaming process and steps involved in delivering content through HTTP using MPEG-DASH, where the overheads and delay trade-offs associated with the HTTP segment duration are analyzed and compared to RTSP.

It is shown that the time delay difference when compared to RTSP is essentially due to the segment duration and buffering on the client side, with a difference of approximately $3 * SegmentDuration$, from which $2 * SegmentDuration$ refers to the buffering time, and $1 * SegmentDuration$ is due to the segment creation time. This formula implies that for a segment duration of 2 seconds the delay difference when compared to RTSP is of about 6 seconds. 6 seconds may not seem much, but in live sports events, for example, it can be a nuisance, especially if other sources of content are providing the same information, such as regular broadcast radio or Over-The-Air (OTA) TV. In spite of the theoretical delay of $3 * SegmentDuration$, in practice, mostly due to caching hierarchies, buffering, and content encryption processes, commercial services typically exhibit live streaming delays ranging from 30s to 65s. Figure 2.19 illustrates the delivery steps leading to delay accumulation.

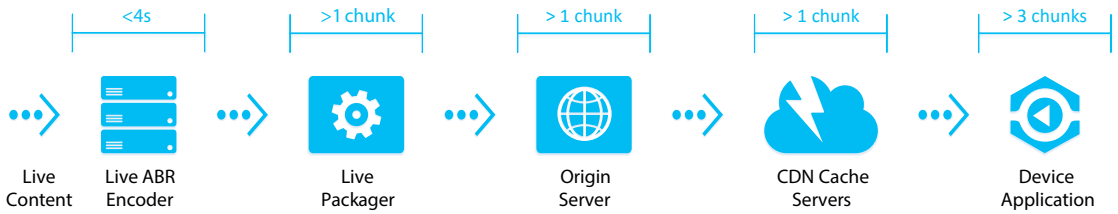


Figure 2.19: Delay Decomposition in HTTP Live Streaming.

An overhead comparison is also performed and is shown to be proportional to the segment duration, though the worst case situation with segment duration of 1 second only implies an overhead of 31Kbps when compared to RTSP's 12Kbps. Increasing the segment's duration to 2 seconds is enough to provide a comparable overhead (17Kbps).

Other authors [196] have looked into the live streaming over HTTP issue from a client adaptation perspective and develop a feedback control loop mathematically formulated that, besides managing the requested stream quality, also adapts the buffer size to the network's conditions. This is done through active link delay measurements using *iperf* [197], which is clearly not viable in production environments, but sets a groundwork for a more scientific / mathematical approach to the adaptation issue that is generally solved using heuristics.

Overall, live content streaming over HTTP is shown to be a viable option, though with some shortcomings, mostly due to the extra delay involved when compared to RTSP, and with developing adequate adaptation algorithms. These are the challenges that should be explored in future technology evolutions.

2.4.8 Conclusion

Multimedia delivery adaptability has been steadily gaining momentum on the OTT panorama as an evolution of traditional streaming technologies. The past years have been marked by the development of several proprietary adaptive streaming protocols, from which Microsoft's and Apple's proposals were the most successful ones, to the point of shaping the current OTT delivery market towards using HAS solutions.

As demonstrated in this section, in spite of solving many issues with older technologies, specifically when faced with varying contextual conditions, HAS presents challenges of its own, notably on client adaptation algorithms, on caching efficiency – which is addressed on Section 2.5 –, and live content streaming.

Despite these open research issues that must be addressed, HAS is the most relevant candidate for a *de facto* standard of next generation multimedia streaming protocols.

Considering its expected dominance in future OTT multimedia networks, it makes sense, then, to consider HAS streaming mechanisms throughout this Thesis' research work, and support them at each step of the overall OTT delivery solution. Due to the available tooling, this Thesis research work uses Microsoft's Smooth Streaming as the reference HAS solution, notwithstanding other competing alternatives which could have been used without implications on the results.

2.5 Multimedia Streaming Caching

The rising popularity of multimedia streaming protocols has lead to its widespread adoption in modern multimedia services, with key representative examples such as YouTube, Netflix, and Amazon Instant Video. Section 2.4 discusses the technical details of this set of streaming protocols at length, and shows that they exhibit very specific traits, which are dependent on the streaming protocol in use.

For example, delivering content using adaptive streaming protocols over HTTP, although conceptually similar to delivering any other web content, presents a set of challenges regarding scalability and maintenance of adequate QoE levels.

CDNs such as the ones described in Section 2.3 are used to address these issues; however, ensuring a good use of available resources while maintaining a high-QoE is challenging, as the performance of CDNs is highly dependent on the nature of the traffic that transverses it and its request patterns; therefore, CDNs must be optimized to take advantage of the characteristics of the content being delivered. As seen in Section 2.3, there are several aspects that impact the performance of a CDN such as the computing resources of CDN clusters, their network interconnections, the request routing systems, and the placement of replica servers; however, in this section emphasis is put into the choice of caching algorithms specifically tailored towards improving the delivery of multimedia content.

Caching algorithms represent a clear example where optimizations to a single component have the potential to greatly benefit the overall solution. Moreover, the other aspects of CDNs are usually harder to modify after the initial deployment, whereas new caching algorithms may be deployed as incremental improvements.

Caches are ubiquitous in modern computing, and have a wide range of applications from low-level caches, at the CPU level, up to worldwide massively distributed implementations. In spite of the very different application scenarios and associated particularities, the cache replacement algorithms used tend to be quite similar.

The general issue of caching has been the subject of extensive research work, ranging from conceptually simple algorithms such as First-In-First-Out (FIFO), LRU, and LFU [198], up to more advanced ones including LRU-K [199], LRU-HOT [200], and Low Inter-reference Regency Set (LIRS) [201]. LRU-K was developed to improve the caching performance of database buffers, while LRU-HOT's target is to keep "hot" items in cache, with the help of backend server-supported content flagging through HTTP MIME type headers, which prevent it from being easily deployed to practical solutions. As for LIRS, it improves LRU for content with weak locality. From these seminal works, Bélády's contribution [202] stands out by providing and demonstrating an optimal caching algorithm (MIN) still used today as a theoretical reference for the upper limit in achievable cache hit-ratios.

In order to better understand how commonly used caching algorithms work, and the benefits that they provide, an overview is conducted on two popular ones: FIFO and LRU, as they are usually the base of other more complex solutions [203].

2.5.1 Reference Caching Algorithms

To exemplify how the sample cache policies work, let us consider a simple reference string: (1,2,3,4,1,2,5,1,2,3,4,5) [204] with a cache size of 3 items, along with FIFO and LRU as widely used cache replacement strategies. FIFO maintains a list of items where the head of the list is the oldest item, and the tail is the latest arrival. This policy removes the oldest ones first. Table 2.2 demonstrates the application of the reference string to a cache employing FIFO, and shows that 3 cache hits are achieved, along with 9 page faults. FIFO is well known for being vulnerable to the Bélády's anomaly [205].

LRU takes into consideration the time of the items' last utilization and removes the least recently accessed items as needed in order to have enough space to insert a new item. As for LFU, it is quite similar to LRU, but instead of bookkeeping the last access time for each item in cache, it stores the number of accesses to the item and removes items that are least frequently used. Table 2.3 demonstrates how LRU's replacement strategy works. With the considered reference string, LRU achieves 2 page hits and 10 page faults.

Iteration	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Request	1	2	3	4	1	2	5	1	2	3	4	5
Result	miss	miss	miss	miss	miss	miss	miss	**hit**	**hit**	miss	miss	**hit**
Page 1	1	1	1	4	4	4	5	5	5	5	5	3
Page 2		2	2	2	1	1	1	1	1	3	3	3
Page 3			3	3	3	2	2	2	2	2	4	4

Table 2.2: FIFO Cache Replacement Policy.

Iteration	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Request	1	2	3	4	1	2	5	1	2	3	4	5
Result	miss	miss	miss	miss	miss	miss	miss	**hit**	**hit**	miss	miss	miss
Page 1	1	1	1	4	4	4	5	5	5	3	3	3
Page 2		2	2	2	1	1	1	1	1	1	4	4
Page 3			3	3	3	2	2	2	2	2	2	5

Table 2.3: LRU Cache Replacement Policy.

The theoretical examples presented in Tables 2.2 and 2.3 demonstrate that the conventional approaches could perform better with respect to their hit-ratios.

Content caches are often a few orders of magnitude smaller in capacity than the corresponding origin servers' storage, which hold entire catalogs of content. Given the capacity constraints involved, one of the main goals of cache replacement algorithms is to utilize as efficiently as possible the available capacity, in order to reduce the load on

upstream origin servers and backend network.

As the performance of caching algorithms is highly dependent on the items' request sequences, a better caching algorithm would have to know which items to keep in cache, in order to maximize the probability of cache hits.

Having established the existing reference caching algorithms and how they work, a literature review of multimedia-specific caching algorithms is performed, starting with the HAS streaming context and its challenges, and ensuing with a review on caching algorithms in the context of multimedia IPTV services.

2.5.2 OTT HTTP Adaptive Streaming Caching

Challenges in Adaptive Streaming Caching

The most popular approaches to adaptive streaming algorithms, relying on Microsoft's Smooth Streaming, Apple HLS, or MPEG-DASH perform two key transformations to the multimedia content: first, they encode it with multiple quality levels, so that the client may select a quality level that suits its particular environmental conditions; second, they break the encoded assets into smaller segments, with a duration of a few seconds.

These characteristics have an immediate impact on the performance of the caching algorithms, as the effective size required to store a full content effectively increases, which in turn has a negative effect on the caching performance, given that the cache size remains constant. Some authors [206, 207] propose the use of real-time transcoding at the proxy-caches in order to reduce the number of quality variants to cache; however, this approach has an unbearable processing cost on large scale delivery networks.

Apart from this immediate side effect of inflating the total size of the *corpus*, i.e. the complete set of content that might be cached, other not so evident challenges arise due to the protocols' adaptive nature, and also because of different video player implementations that rely on track selection heuristics that have an impact on caches' performances.

One common problem is that of *bit rate oscillation* [208, 209], which manifests itself in the form of repeated cycles where the playback client adaptation algorithm overestimates the available content bandwidth when the requested segment is in cache (*cache hit*), and then proceeds to optimistically request a subsequent segment with a high quality level that is not in cache (*cache miss*), thus being subject to an additional delay, that might force the client to review its link quality estimates and request another segment of lower quality. This loop may perpetuate itself, with nefarious consequences to users' QoE and cache performance.

In addition to these issues, caching algorithms relying on content pre-fetching to improve their performance also suffer when used with adaptive streaming protocols, given that a decision must be performed on the quality levels that should be pre-fetched, or face the impact of pre-fetching every available quality level at the expense of cache storage [210, 208].

Finally, because adaptive streaming mechanisms have been developed with QoE improvement in mind, the caching algorithms should not hinder their efforts in providing

the best possible experience to the end user, and facilitate QoE maximization. Most approaches for QoE optimization in caches rely on fundamental characteristics of the content demand patterns, such as minimizing the initial playback delay, by prioritizing the initial segments of video content, or avoiding large variations in access latency [211, 212].

HTTP Adaptive Streaming Caching Algorithms

The authors of [213] propose *DASCache* with the goal of handling heterogeneous caching of video content on top of ICNs. Their approach focuses on minimizing the average *access time for bit* of the requested content, with the ultimate purpose of improving users' QoE. To that end, *DASCache* works periodically, by monitoring the incoming requests to collect usage statistics in order to predict the bit rate that the users' requests will need, and then placing the forecast content in the caches, at least for the duration of one period. The content placement procedure is an optimization problem, which is solved using binary integer programming.

The authors demonstrate that, when compared with Periodic-LRU and Periodic-LFU, which only perform purging decisions at the end of a given period, *DASCache* performs significantly better.

A different approach is pursued in [206], where the issue of bit rate oscillation, and cache under-performance due to client adaptation algorithms is addressed. In order to address the caching inefficiencies and the impact of bit rate oscillation, two modifications are performed.

First, the DASH MPD is modified to include information regarding already cached items, so that the client may be aware that some segments already exist in the server's cache, while other would have to be fetched from a remote location. This feature will naturally require a modification of the client adaptation algorithms to take this extra information into consideration, which may not be feasible to deploy.

The second modification tries to address the issue of item size amplification, caused by having to cache multiple versions of the same segment, with different bit rates. A solution is proposed that relies on on-the-fly transcoding to downscale high quality items into lower quality items that the clients request. Although it is argued that line speed transcoding is feasible, this solution may not be cost effective.

In [208], Video Shaping Intelligent Cache (ViSIC) is proposed as video aware cache server implementation, with the main goal of solving the bit rate oscillation issue. The approach relies on traffic shaping to ensure that the client adaptation algorithms do not make too optimistic decisions with respect to the path bandwidth. The influence of traffic shaping on the clients' adaptation algorithms, in addition to stabilizing the bit rate oscillations, has a positive impact on the caches, given that the cache server now has the possibility to avoid using certain segment alternatives that are not in cache in favor of others that are, therefore increasing the cache hit-ratio.

Naturally, this implementation requires that the server is fully aware of the adaptive streaming protocol in use, and that it is able to process the manifest files (MPD) in order understand which bit rates are available so that adequate decisions on content bit

rate selection may be performed. The simulation results indicate that ViSIC is able to provide smooth video playback sessions, without bit rate oscillations, while being able to accommodate variations in the available path bandwidth.

Instead of relying on reactive caching algorithms, the authors of [210] propose a proactive attitude towards HAS, using pre-fetching. One of the main goals of their solution is to offset some of the traffic expected in prime time into other time windows, so that a reduction in peak upstream bandwidth utilization is observed. This argument is based on the residual cost of pre-fetching traffic in off-peak hours, when compared with the impact of peak bandwidth utilization at certain times of the day. The work focused on YouTube consumption data, using DASH streams, and relies on the use of a centralized cache coordination agent. An estimated reduction of 20% of peak upstream traffic is shown.

All of the previously addressed caching strategies have the underlying purpose of improving caching efficiency, which ultimately benefits the end users' QoE; however, they do not explicitly focus on clients' QoE. [211] on the other hand, approaches the adaptive streaming caching perspective from a QoE optimization approach in the specific context of wireless content delivery. To that end, a logarithm model for QoE is derived from experimental results, and a snapshot optimization problem is formulated that aims to discover the subset of items to cache that maximizes the users' QoE, under the assumption that the cache might reply to a request for a particular bit rate content with a different one of a similar, acceptable, bit rate.

This flexibility in the bit-rate provided significantly improves the cache's performance, as its effect is similar to that of increasing the effective cache size.

Another work that focuses specifically on optimizing the users' QoE through caching policies is that of [212]. The authors propose the deployment of proxy-caches to the users' local Access Points (APs), which are responsible for transparently providing a content cache and fetching the requested content from backbone servers. Because this approach targets wireless environments, where the AP has full knowledge of the link quality to its clients, the proposed solution uses this information as a cache metric, so that assets with a higher probability of being requested – due to their bit rate – are kept in cache. By complementing this mechanism with pre-fetching, based on the same rationale. The results demonstrate an improvement in QoE, a significant reduction in freezes, a higher overall channel data rate, and a reduced QoE standard deviation.

2.5.3 Caching In IPTV Multimedia Services

The popularity and ubiquity of IPTV multimedia services make it an appealing use case of multimedia streaming caching algorithms, which has been the subject of extensive research works, and is now in a migration process from managed delivery services to an OTT approach. A particularly popular group of multimedia services in IPTV is that of *timeshift* TV, i.e. a functionality that enables nonlinear access to previously aired TV content, as is the case of *Pause TV*, *Restart TV*, *DVR*, and *Catch-up TV*. To cope with a growing demand for these unicast timeshift services, operators deploy caches in strategic locations, to reduce the impact on core and aggregation networks.

In spite of a large research body encompassing caching issues in many areas, there are a limited number of research studies that address Catch-up TV content caching; therefore, this section presents a literature review on caching algorithms specifically tailored towards these rich services.

While many aspects of CDN optimization are not directly dependent on the nature of the content being served, as they are generally built in a content-agnostic manner, the application of "standard" CDNs to multimedia streaming delivery and, in particular, to Catch-up TV delivery is far from optimal, as this type of content exhibits a dynamic demand behavior that is not properly accommodated by traditional CDN caching algorithms [139, 52]. Improving caching performance requires taking into consideration the underlying content demand patterns, and properly exploring them.

The work of [214] stresses the big challenge of Catch-up TV caching, and investigates caching strategies suitable for this service. To that end, a model is built that takes into account the evolution of content popularity, which is used by a caching algorithm that keeps track of the requests per item and dynamically builds the said model to estimate the relative importance of items and make caching decisions. The results show that this approach is able to outperform LRU and LFU for the 1.640 traces tested; however, the impact of the dynamic model building overhead is not considered in the simulations.

Borst *et al.* [215] tackle this issue from a cooperative approach, taking inspiration from distributed file systems and large scale information systems, but focusing on bandwidth savings instead of latency reductions, which are assumed to be far less important in these contexts. This study finds that pro-actively loading content into cache nodes incurs a significant cost penalty, hence suggesting that content should be reactively cached. Additionally, its conclusions also indicate that N-tier caching should be limited to 2 layers, as it simplifies implementation and management, while retaining the vast majority of caching benefits. After formulating the problem using Integer Linear Programming (ILP), the authors proceed to propose cooperation algorithms which are shown to obtain results close to optimal, under the assumption of perfectly known and modeled content popularities.

In [216], a predictive approach is taken towards content popularity, where data traces are used to fit synthetic Gaussian, exponential, and power law models which are then used in a modified LFU caching policy. This technique is shown to outperform the basic LRU algorithm; however, because it assumes historical knowledge on each multimedia item being cached, it is not viable in operational environments where new content is added every day.

A complementary work is performed in [217], where Abrahamsson *et al.* provide an empirical IPTV work model based on a realistic simulation which considers the large discrepancies in content popularity, with the purpose of evaluating the performance of traditional caching algorithms, including LRU and LFU, and estimating the bandwidth requirements of time-shift services. The study's conclusions demonstrate that LFU is the most favorable caching approach; however, the study neglects the fact that Catch-up TV content has a life-time expectancy that must be taken into account, so that popular content that is no longer valid does not prevent new content from populating the caches.

Another study is conducted in [218] regarding a *TV-on-Demand* service providing Catch-up TV, T-VoD, and S-VoD content. In spite of the mixed service-type analysis, this study's conclusions support the occurrence of the Pareto-principle, or the 80-20 rule, whereby the 20% most popular assets are responsible for 80% of the total content requests. Research is conducted on the content *cacheability*, which is shown to be very high even when using traditional caching algorithms such as LRU and LFU.

This research work is improved in [139], where additional effects are exploited, such as program popularity variability with time. A characterization of its decay with time and genre is also provided. The results show that the content genre and the Catch-up TV availability window plays a very important role on the performance of caching algorithms and on the streaming bandwidth required from the origin servers.

The work in [52] focuses on prefetching content to the clients' devices. Caching and prefetching are highly intertwined as both require a deep understanding of the contents' characteristics, and work towards the goal of reducing peak bandwidth consumption. Several conclusions are withdrawn regarding how users behave: in addition to showing a high engagement, users access the service in time-spread manner throughout the day, and exhibit strong preferences for a small set of programs.

2.5.4 Conclusion

Caching is, by itself, a challenging proposition that has been widely researched and that has a very significant impact on the overall performance of any data retrieval system. When applied to the context of OTT HTTP Adaptive Streaming, a set of additional challenges arise, partly due to the client adaptation schemes, and to the increase in overall content size caused by the multiple bit rate encodings of the available assets. Examples of these challenges include dealing with bit rate oscillation, pre-fetching, and QoE maximization.

In order to address these problems, a set of HAS-specific caching algorithms have been proposed; however, most focus on very specific problems that require full domain knowledge, or excessive computing resource, thus suffering from general applicability.

With respect to the particular case of IPTV multimedia services, which are migrating to OTT HAS scenarios, additional issues arise, specially because of the dynamic content popularities that make it hard for caches to operate efficiently.

The issue of caching in multimedia systems is full of challenging propositions that must be addressed by next-generation OTT multimedia CDNs. In the context of this Thesis' work, research is conducted to improve the performance of caching algorithms on CDN nodes using a novel approach that models content demand to add content-specific knowledge to caching layers, enabling significant performance gains when compared to competing solutions.

2.6 Quality-of-Experience (QoE) on OTT Video Networks

There has been a growing scientific interest in QoE over the last decade motivated by the multi-dimensional characteristics of the human experience in technology interaction [219]. However, common scientific approaches tend to ultimately focus on QoS metrics under the assumption that mere improvements in QoS will lead to improved QoE, disregarding user-related aspects such as expectations, or previous experiences.

This perspective has been supported by the latest technological advancements in content delivery technologies, tools for developers, and access networks (such as FTTH, LTE, etc), which usually improve the overall QoS and, to some extent, the QoE. Nevertheless, it is a well-know fact [220] that QoE is the key factor that should be looked into, as users have high-quality expectations that, if not met, might jeopardize their loyalty.

QoE is defined by International Telegraph Union Telecommunication Standardization Sector (ITU-T) [221] as:

> *The overall acceptability of an application or service, as perceived subjectively by the end-user.*

There is a departure from the traditional QoS perspective which only encompasses the networking aspects of the services such as [220, 222]: *throughput*; *goodput*; *delay*; and *loss ratio*. The description of QoS, according to ITU-T, is:

> *The totality of characteristics of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service.*

Kilkki [220] clearly differentiated the two concepts with the following statement:

> *It is quite meaningless to say that the goal of network operations is high QoS (a similar statement would be to claim that the purpose of life is to speak perfect English).*

A user performing web-browsing is not concerned with the loss ratio of the connection, he only cares about opening the web-page he was looking for in a reasonable amount of time. Figure 2.20 illustrates the different scopes of QoE vs QoS. QoE reflects a different perspective on quality monitoring and tries to answer the *why* question: *why* is the video stuttering? *why* does the user feel frustrated? [223].

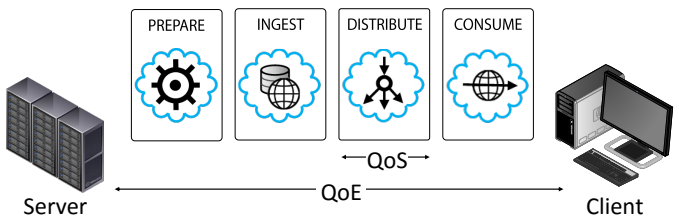


Figure 2.20: QoE vs. QoS Scope.

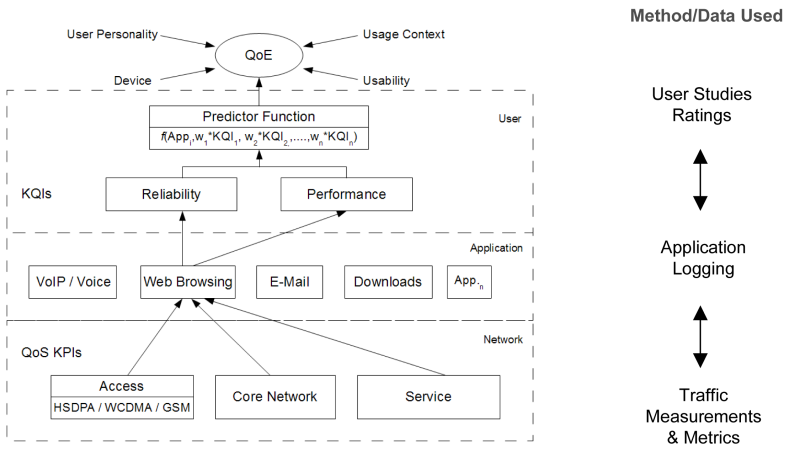


Figure 2.21: Framework for modeling the QoE of networked services. The application/service specific QoE predictor function is derived from linking performance indicators from three different layers, i.e. network, application and user [4].

Figure 2.21 is taken from [4], where the logarithmic laws in quality perception and the complex relationship between factors that ultimately culminate in the user's QoE are evaluated. There are external factors such as users' personalities, the application/service usability, and the usage context, to name a few; and then there are service-related Key Quality Indicators (KQIs) that also influence the users' QoE.

It is crucial to understand which factors can be controlled and which cannot in order maximize the users' QoE, but in order to do so, there must be a way to measure QoE, as something that cannot be measured, cannot be optimized. QoE measurement, or assessment is divided into two main categories, depending on how it is performed: *subjective* and *objective*.

Subjective QoE assessments involve surveys to people that have been exposed to the service or application whose QoE is being assessed. These surveys rely on users' *opinions* to rate the service performing under different conditions. The rating system may be based on *qualitative* or *quantitative* evaluations.

Qualitative evaluations tend to focus on comparative evaluations between different experiments, such as indicating that the first experience was more pleasant than the second one. As for *quantitative* assessments, users are asked to use a number to grade their experience according to pre-established scales; thus, being objective. The latter is more widely used as it facilitates data processing. In the context of video reproduction, International Telegraph Union Radiocommunication Sector (ITU-R) recommends the usage of BT.500–13 [224] for video quality assessment.

In contrast to the previously described subjective assessment methods, which are laborious, time-consuming and expensive, the objective approach is an alternative that builds on the technical characteristics usually provided by QoS parameters to model and extrapolate a perceptual quality indicator. Because there is usually no way of ensuring that, by itself, the model is accurate, objective assessment algorithms are usually trained

and modeled by resorting to known QoE estimations from subjective models.

Naturally, how objective QoE assessments are performed depends strongly on the service under consideration. In the case of uncontrolled OTT networks, there are QoS and QoE-specific metrics that may be considered when focusing on OTT video streaming.

The following subsections will focus on the main QoE-influencing factors, starting with a general perspective on QoE in the context of video reproduction, followed by more specific analysis of QoE in adaptive streaming systems (DASH in particular).

2.6.1 QoE in Video Reproduction

QoE is a subjective metric; hence, most of the video quality assessment studies rely heavily on subjective quality assessment methods. The main metric used to rate the quality of a video is Mean Opinion Score (MOS), where users are asked to rate their experience in a 5 point scale. Given the subjective nature of this metric, the tests must be performed in controlled environments [225].

In addition to the subjective approach to QoE measurement, there are some objective metrics that may be used, such as video frame rate, resolution, and compression level, which can be monitored using specific tools [226], and then correlated with the users' perceived quality. These correlations enable the creation of QoE models based on technical parameters.

There are two commonly used types of models, depending on the dimensions considered: pixel-domain models, and bit stream-domain models.

The pixel domain models may be further subdivided into 3 main categories, according to the information required by the models' algorithms:

- Full-Reference Models - require the complete original reference video for comparison; they provide high accuracy and repeatability at the expense of intense processing and/or bandwidth;
- Reduced-Reference Models - require a partial view on the reference video, use features extracted from the original video to perform the comparison. Trades-off bandwidth for the reference signal with measurement accuracy;
- No-Reference Models - rely only on the degraded signal to perform a quality estimation; hence, the estimates are less accurate. The reference signal is unknown.

The second type of models - bit stream models - inspect the video flow and use the extracted parameters to infer QoE. Relevant parameters include: flow bit rate, packet losses, jitter, and RTT. In these models, the video is not effectively decoded.

As far as video image quality is concerned, popular metrics using full reference models include PSNR, and Mean Squared Error (MSE); however, they serve merely as indicators given that under some particular circumstances the results may be deceiving [225]. Nevertheless, on most situations they do provide valuable insight on the image quality.

When transmitting video over TCP/IP, using progressive streaming for example, other factors must be taken into account that influence QoE.

Due to the reliable nature of TCP, the issue of lost frames does not present itself

as it does in on UDP based streaming; thus, the variations of QoE in this scenario are usually related to network delays and buffering issues [227]. If the buffer is being filled at a slower pace than it's being consumed, the playback will frequently have to stop and wait for more data. Because these stops have a large impact on the perceived QoE, the authors of [227] have proposed a focus on the temporal structure of the videos being streamed, which culminated into 3 main metrics:

- Inital buffering delay - Time delay until initial playback is started;
- Average rebuffering duration - How long rebuffering events last;
- Rebuffering frequency - How often rebuffering events occur;

Similar conclusions were drawn in [228], where an aggregate metric called *pause intensity* combines both the average rebuffering duration and its frequency. The results attained showed a remarkable correlation between pause intensity and MOS variability.

It is also shown in [227] that the initial buffering delay does not have a large impact in the QoE, the users would rather wait a bit more for the playback to start, than have rebuffering events throughout the playback session. Although this is a valid assumption on VoD services such as Netflix, in the context of live broadcast streaming, the users are more sensitive to initial-buffering delays, as they expect the stream to start right away.

Both studies also concluded that in the particular context of streaming over TCP/IP, the temporal aspects of the video playback have a higher impact on QoE than spatial artifacts, such as macroblocks, or blurriness.

2.6.2 QoE in Adaptive Streaming

Adaptive streaming technologies aim to improve the QoE of the streamed video over time by relinquishing some degrees of control to the end client, which may then adapt to changing conditions and minimize rebuffering events.

The client is in a unique position to assess its environment conditions and must be able to decide which stream to consume, from a set of server-provided alternative streams, each with different video and/or audio characteristics. To that end, several parameters must be modeled, estimated, and monitored, such as:

- *Network Resources* - Bandwidth, Delay, Jitter, Availability ;
- *Capabilities* - Available Memory, CPU, Screen Resolution, Remaining Power ;
- *Streaming conditions* - Buffer Size, Desired seek speed, Desired start-up delay.

These extra degrees of control add to the number of dimensions contributing to a good user QoE, and despite having the potential to ultimately benefit QoE, they may very well hinder it if the client control algorithms are not adequately tuned.

Scenarios where there is "enough bandwidth, but not enough CPU power for decoding high-bitrates", or there is "good bandwidth, available CPU and memory, but low remaining battery power", are commonplace, and must be accounted for.

An extensive QoE study is performed in [229] where it is shown that the crucial QoE advantage provided by HAS when compared to progressive streaming is mostly

due to the reduction of rebuffering events or stalls. This metric has been shown to have a critical impact on the MOS. In addition to these findings, relationships are also established between the QoE and factors like quality switching frequency, initial playout delay, startup bit rate and average bit rate. All of these factors must be weighted in order to maximize the users' QoE.

Due to the novelty of this technology, when compared to the more traditional push-based adaptive streaming techniques, several challenges and opportunities arise. One of the crucial-for-success challenge is the development of adequate methodologies and metrics for assessing the users' QoE for adaptive streaming services.

Realizing this need, both 3GPP [230] and MPEG [22] bodies identified QoE metrics for DASH, which apply to adaptive streaming technologies in general. The 3GPP proposal also specifies methods for QoE reporting back to the network servers which may provide crucial insights.

Monitoring the QoE is highly beneficial for debugging failures, managing streaming performance, improving client adaptation technologies, and also to provide valuable input to resource provisioning systems.

QoE in 3GPP DASH

3GPP and MPEG identified QoE performance metrics and reporting protocols as playing a critical role in optimizing the delivery of Adaptive Streaming services, and have thus considered them in their DASH specification.

3GPP's TS 26.247 [230] specification is quite detailed and includes mechanisms for triggering client-side QoE measurements along with the specification of protocols for reporting them back to the server. 3GPP mandates that client devices supporting QoE features (an optional requirement) have to support the full set of the requested metrics.

The QoE reporting feature is mainly comprised of three stages. In the first one, the trigger phase, the server requests QoE reports from the clients by using either the MPD or the OMA Device Management (DM) QoE Management Object to specify a percentage of clients that should activate the QoE reporting features. The clients will then use a local random number generator to decide whether they fall into the specified percentage of devices that should report the metrics.

The next phase regards the actual gathering of QoE information, which happens according to the configuration specified in the MPD or OMA DM.

Finally, the client reports the metrics back to a network server.

An extensive amount of metrics are collected so that the servers monitoring the QoE of their clients are able to accurately estimate the users' QoE. Although the focus of this section is on Adaptive Streaming, 3GPP's TS 26.247 also specifies a subset of QoE metrics that should be used in the event of progressive streaming sessions.

Regarding the specified Adaptive Streaming QoE metrics, they are as follows:

1. **HTTP Request/Response Transactions**

   The client must provide a list of all HTTP requests and responses finished within the QoE metric collection period, specifying the following metrics:

(a) Type of request, (MPD, MediaSegment, ...);
(b) Request url and actual url if any redirect was performed;
(c) HTTP response code;
(d) *byte-range-spec* part of the HTTP Range header;
(e) Request timing information (time at which the request was sent and the response received);
(f) Throughput trace information for successful requests.

2. **Representation Switch Events**

   A switch event is triggered when the first HTTP request for a new representation is sent. It represents a client decision on the representation that should be reproduced.

   (a) Time of switch event;
   (b) Media time of the earliest media sample played out from the new representation;
   (c) Representation Id;
   (d) SubRepresentation Level.

3. **Average Throughput**

   Report of the average throughput observed by the client during the measurement interval.

   (a) Total number of bytes in the body of HTTP responses received;
   (b) Activity time in milliseconds (i.e. excluding inactivity periods);
   (c) Start time of the measurement interval;
   (d) Measurement duration;
   (e) Access Bearer for the TCP connection for which the average throughput is reported;
   (f) Inactivity type (pause, buffering, ...) if known and consistent in the report period.

4. **Initial Playout Delay**

   The initial playout delay is considered to be the time elapsed between fetching the first media segment and retrieving that segment from the client buffer for playback.

5. **Buffer Level**

   Reports a list of buffer level status events measured during playout at normal speed.

   (a) Time of the measurement;
   (b) Buffer level in milliseconds.

6. **Play List**

   Contains a list of playback periods. A playback period is defined as the time interval between a given user action and whichever occurs soonest: a subsequent user action; the end of playback; or a failure that stops playback.

(a) Timestamp of the user action that triggered the playback period;
(b) The media (presentation) time at which the playout was requested;
(c) The action type that triggered the playout period (initial playback, seek, resume, user requested quality change, ...);
(d) Trace of played segments, containing their RepresentationId, SubRepresentation level, timing, playback speed, stop reason, and duration. The trace may contain entries for different representations that overlap in time, due to different representations being played simultaneously (e.g. audio and video).

7. **MPD Information**

   In order to provide adequate information to servers that may not have access to the MPD, this information must be sent whenever any other metric references a Representation for which MPD information has not been reported yet, so that the servers have sufficient information on the media characteristics.

   (a) Representation Id addressed by the QoE metrics report;
   (b) SubRepresentation level addressed. If not present, the report concerns the complete representation;
   (c) MPD Info. Complete MPD Information for the specified Representation/-SubRepresentation level.

After gathering the required metrics, the client then compiles the QoE report in an XML format, complying with the schema defined by 3GPP, and sends it to the server using a simple HTTP POST request.

2.6.3 QoE Estimation on HTTP Adaptive Streaming

Understanding how QoE may be estimated and how it can be improved along the content distribution pipeline is of paramount importance; therefore, a literature analysis is required on methods for estimating the QoE.

Performing a subjective study of QoE for a given video playback session is costly and cannot be performed in real time; thus, automatic methods for estimating QoE are valuable and desirable. The authors of [231] realized this issue and proposed an adaptation of Pseudo-Subjective Quality Assessment (PSQA) [232] to HAS using H.264/AVC. The created no-reference QoE estimation module based on Random Neural Networks (RNNs) was able to provide fair estimates on the QoE of 18 validation videos. However, some restrictions were imposed, namely on the analyzed quality dimensions, as the estimation module was limited to the Quantization Parameter, used as an indicator of video compression and bit rate, and on a model for playout interruptions.

Despite being able to capture important metrics and conveying them into a QoE estimate, this QoE estimation module fails to encompass other metrics identified as playing a significant role on HAS QoE such as quality switching, playback resolution, or even the initial playout delay [229].

In another study [233] the issue of QoE estimation was addressed in the context of Radio Access Networks (RANs). Two different base video clips were prepared to target

tablets (iPads) and smartphones (iPhones). Using these base video clips as HLS sources, several LTE network conditions were simulated to create different playback scenarios. The resulting video clips were reconstructed at the client devices in order to allow for offline and comparable MOS evaluations for each of the 90 reconstructed clips, which were later evaluated by a total of 500 volunteers.

The subjective evaluations were then used as the ground truth for MOS linear prediction models where the quality metrics considered were PSNR, Structural Similarity (SSIM), nominal chunk bit rate, and chunk-MOS - where a given MOS is associated with a specific chunk quality level.

The model's result show that the bit rate based model is the worst, which can be explained by the non-linear relationship between bit rate and MOS. The PSNR and SSIM approaches provided fair results, while chunk-MOS's performance was shown to be the best, especially with regard to sensitivity to non-optimal model parameters. These characteristic make the chunk-MOS approach suitable for classifying unseen content.

The authors state that the impact of quality level switches is not significant in their test, even though other studies [229] clearly demonstrate its impact on QoE.

A different approach is taken in [234], where the authors of Quality Monitoring (QMON) take on the QoE estimation issue with a network-monitoring approach relying on Deep Packet Inspection (DPI), by placing an intermediate buffer proxy between the video source and the client. Given the network proxy approach, no direct access to the client devices is required. Despite being focused on progressive streaming, and not addressing HAS directly, a method for extending QMON to support HAS is suggested.

The MOS estimation relies solely on a buffer stalling evaluation. The buffer stalls are weighted in a "negative impact" exponential function that aims to capture the aggravated impact on QoE at each subsequent stall event, along with the duration of each stall. The network monitor relies on buffer fill level estimations based on timestamps embedded on the video payload of a given TCP flow and has 3 modes of operation.

The first one, the *exact method*, decodes every TCP packet to try and extract the video timestamp and compare it with the timestamp of the respective TCP segment. Because every single packet is decoded, this is the most accurate method, at the expense of intense computational requirements.

As for the second approach, the *estimation method*, increases the processing speed of QMON by fully decoding only the video header and extracting the size and duration of all subparts, which are then used as baselines for estimating the buffer fill level relying solely on the amount of TCP data streamed. This is a fair approach if the network does not experience a significant number of TCP retransmissions, as these add up to the amount of data streamed and influence the buffer fill level estimation.

Lastly, the *combined method* uses the previous methods dynamically to adapt to the experienced transport conditions. If a significant amount of TCP retransmissions is experienced, the *exact method* is used, otherwise the *estimation method* is preferred.

The performance evaluation of the buffer fill level estimation is shown to be accurate on both the exact and combined method, with the estimation method falling behind.

Regarding the MOS estimation, which is the focus of the paper, despite having

provided a proposal for an estimation metric, the provided metric is not validated with any test subjects; thus, its accuracy is questionable.

In [5], a Time-Varying Subjective Quality (TVSQ) computation is performed relying on the videos' estimated Short-Time Subjective Quality (STSQ). As the authors put it, TVSQ *"is a time series or temporal record of viewers' judgments of the quality of the video as it is being played and viewed"*, while STSQ is a *"scalar prediction of viewers' subjective judgment of a short video's overall perceptual quality"*. The dynamic model is defined to consider the quality-variation aspects of HAS in such a manner that online QoE evaluation of test videos is feasible.

This approach stems from the fact that video watching is a time-varying experience, with a non-linear relationship between the current frame quality and the current experience quality, due to the viewers' memory effect.

In order to create the STSQ predictions, the Video-RRED [235] algorithm is used due to its good prediction accuracy and performance. Then, the STSQ inputs are fed into a dynamic system model of TVSQ, which then outputs a prediction of TVSQ.

The TVSQ system proposed is an Hammerstein-Wiener (HW) model with generalized sigmoid input and output functions capable of capturing the non-linearities between the input STSQs and the output TVSQ, and an intermediate linear Infinite Impulse Response (IIR) filter, as shown in Figure 2.22. The model's parameters are then estimated with the help of reference TVSQs generated from subjective video-viewing tests.

In the validation phase, the "leave-one-out" cross-validation approach is taken, i.e. all the reference videos except one are used to train the model, and the model is then applied to the validation video. The results show a high linear and Spearman's rank correlation coefficient (~ 0.9) between the measured TVSQ and the estimated one, and appear to be robust for the tested data set. With regard to the stability of online predictions of TVSQ, the model is shown to be stable in the presence of untrained videos.

The proposed TVSQ-based approach greatly succeeds in modeling non-linearities and the chunk-base nature of HAS, providing a valuable contribution to the estimation of QoE in HAS sessions. Some aspects such as initial playout delay are not addressed, and the practical feasibility may be not be ideal given the STSQ estimation approach taken, however, with adjustments to the way how the STSQs are computed, this model demonstrates an excellent potential for accurate, online, QoE estimation.

Figure 2.22: Hammerstein-Wiener model for TVSQ prediction [5].

An alternative the previous model is considered in [236]. The authors focus on the DASH variant of HAS, and identify a set of key QoE-impacting metrics. The buffer

overflow/underflow events are analyzed with respect to their re-buffering frequency and average duration; the frequency and amplitude of quality switches are also considered, given their proven impact on QoE [237]; and, lastly, an objective content quality level metric is examined which encompasses factors such as video bit rate, frame rate and quantization parameter.

Taking the previous parameters into consideration, An eMOS (estimated MOS) analytical exponential model is developed and is later calibrated through subjective testing. The model is presented in equation 2.1, where the $\{a_0...a_{N-1}\}$ and $\{k_0...k_{N-1}\}$ parameters represent the weights associated with each metric $\{x_0...x_{N-1}\}$.

$$eMOS = \sum_{i=0}^{N-1} a_i * x_i^{k_i} \tag{2.1}$$

After subjective testing calibration, the model is shown to provide an adequate performance and closely tracks the users' perceived QoE. The proposed approach, while failing to encompass memory effects in QoE such as the ones modeled in [5], succeeds in providing an eMOS model able to encompass the main QoE-impacting factors in HAS (buffer and quality switches characterization plus content quality modeling).

2.6.4 QoE Optimization on HTTP Adaptive Streaming

Given the broad scope of parameters that affect the final QoE, there are a multitude of aspects that can be optimized in order to increase the overall QoE. The QoE optimization aspects in HAS may be broadly subdivided into three categories (Figure 2.23):

1. *Content preparation* - Aspects related to how the video is encoded and segmented, e.g.: codec; codec parameters; and segment duration;

2. *Delivery* - Factors intervening in the content delivery process. These may encompass the use of proxies, caches, and network-specific optimizations, to name a few;

3. *Consumption* - In HAS, client adaptation algorithms play a very significant role in the final QoE. This category encompasses optimizations performed at the client side;

The ensuing sections provide a literature review on these three different categories.

Optimization through content preparation

The content preparation process is crucial on any HAS technology as the process is performed only once but the content is consumed multiple times; thus, any optimization performed at this stage is transparently reflected across the overall HAS solution.

The two dimensions on which optimization is usually performed are encoding and segmentation. In the encoding dimension, different codec parameters may be adjusted

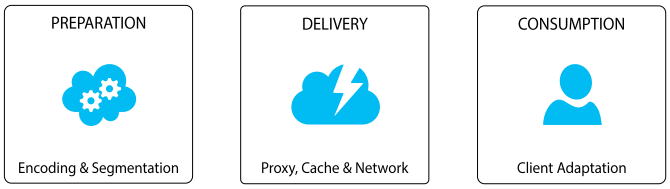


Figure 2.23: QoE optimization aspects on HTTP Adaptive Streaming.

such as the codec in use, frame rate, resolution, and Group of Pictures (GOP) size to name a few. As for the segmentation, the segment size dimension plays a significant role on the content buffering and delivery aspects.

The authors of [238] focus on the first aspect of content preparation optimization: encoding. Specifically, on optimizing the encoding process to reduce the streaming bit rate required on adaptive streaming platforms, where the content encoding process should be aware of the segmented nature of the content.

In most commercial H.264 [174] encoding solutions each segment starts with Intra-coded pictures (I-frames), i.e. fully specified pictures, while the other frames are coded with P-frames (predictive-frames). This structure, denoted IPPP, simplifies encoder requirements given that the encoder does not need to be aware of the video content, it just has to generate I-frames at a regular interval. As a side-effect, the I-frame placement is not optimal, and the Rate–distortion (RD) performance is reduced.

To optimize the encoding process, the authors propose a solution based on scene cuts which are used as segment boundaries; thus, the segment size depends on the encoding process instead of being fixed at the typical 2 or 10 seconds. These optimizations are shown to allow for a reduction of about 10% in bandwidth for a given RD target.

In addition to the impact on RD performance, the segment size plays an important role in other aspects of end-to-end HAS solutions. Smaller segments allow for quicker client-adaptation, and lower buffering delays (which may be important in live streaming, for instance), but may present a higher overhead on different parts of HAS technologies, such as the MPD (or manifest) size - smaller chunks translate into an increase on the total number of chunks, and on larger MPDs being required to describe them.

Optimizing the delivery process

A properly optimized delivery solution is a requirement to ensure that the client-requested content is delivered in a timely and scalable fashion. Given that OTT networks cannot provide any performance guarantees by themselves and span different access technologies (such as radio, Ethernet, and fiber), mechanisms are required to cope with this uncertainty while providing a good QoE to the users.

In [239], the issue of maximizing QoE in LTE wireless networks is considered. Delivering media over wireless links is a well known challenge, as the shared medium is typically the bottleneck [240] due to its resource availability constraints and high variability. In these networks, each user consuming media with an HAS player will adapt

individually to the resources allocated by the scheduler in the eNodeB. However, this scheduler is not content-aware: it merely contemplates channel conditions to perform the scheduling decisions.

As a method for maximizing the overall QoE, a modified DASH server, along with a proxy placed at the base station is suggested. Due to the connection of the proxy server to the eNodeB scheduler, the proxy server is able to gather information on the bandwidth allocated to each client, and performs request rewriting off the client's segment requests. This approach ensures that a client never requests content with a bit rate larger than the network can handle, avoiding playback stalls, quality up-shifts that are not stable, and allows for an adequate bit rate selection at the beginning of playback, instead of starting with the lowest quality representation. These approaches are shown to improve the QoE MOS by at least 0.4 (in a 1 to 4.5 scale).

While this is an interesting approach, its feasibility may be somewhat limited given that the request-rewriting proxy requires direct information from the eNodeB scheduler.

Another common approach to optimize the delivery process is to focus on caching mechanisms, such as the one depicted in [211], where a QoE-driven cache management system for HAS over wireless networks is presented. The authors focus on optimizing the overall users' QoE, given a set of playback bitrates, and cache size constraints. Particularly, the cache is populated with chunks from a set of playback rates that are known to maximize the overall QoE (considering every user), and subsequent requests for chunks from the clients are rewritten in order to supply them with the closest representation of the requested bit rate. This provides substantial statistical gains to the caching engine, given that under a particular storage budget a larger number of media items are reused.

Other authors, such as [241] approach the issue from an in-network perspective. The focus of the article is on granting telecom operators a manner of maximizing revenue for on-demand streaming, by allowing the prioritisation of users with higher subscription levels on managed networks. It is intended that "premium" users maintain their 'premium' QoE service, at the expense of QoE of other users sharing the same link.

The problem is defined and solved with an ILP approach formulated to provide access to all users, while maximizing the client's utility relative to its subscription level. The results show that, in addition to maximizing the overall utility (w.r.t. to the subscription level), a noticeable side effect is attained: because each client is eventually restricted to a set of tracks so that the overall link capacity is respected, there is a markable reduction in the number of bit rate switches of the clients, which in turn leads to an improvement of user's QoE.

Optimizing client adaptation mechanisms

The remaining optimization category falls on the client adaptation mechanisms. The client heuristics play a significant role in estimating the adequate chunk that should be requested to the HAS server. The client must take into consideration factors such as: screen resolution, content frame rate, chunk error rates, bandwidth estimation, and buffer management, to name a few. The interplay of all of these heuristics determine the client behavior, and ultimately the user QoE. Recent studies [242, 243, 244] have

shown that current commercial implementations of HAS players present several issues in maximizing the usage of the available network resources, in providing a smooth playback, and in assuring fairness between competing clients.

An excellent evaluation of optimal adaptation trajectories is performed in [245]. The authors create an analytical model of the optimization issue as a Multiple-Choice Nested Knapsack Problem (MCNKP) problem, which is then used as a baseline comparison for the performance of client adaptation algorithms.

In the evaluation section, it is shown that their implementation a DASH plugin [246] for VLC [182], based on a Proportional-Integral-Derivative (PID)-like controller is able to significantly outperform Microsoft's Smooth Streaming player in a number of benchmarks, such as: rebuffering time, average bit rate, buffer size requirements, and number of quality switches. With regard to fairness in the presence of multiple clients sharing the same bottleneck link, the solutions are shown to perform similarly.

Jiang et al. [244] approaches the client adaptation issue from 3 main perspectives: *Fairness*, so that competing clients sharing a bottleneck link get a fair share of network resources; *Efficiency*, to utilize the most out of the available resources; and *Stability* to prevent unnecessary bit rate switches that my affect the users' QoE.

The issue of chunk download scheduling is carefully analyzed in order to demonstrate that current players relying on periodic download schedulers fail to provide accurate bandwidth estimation in the presence of multiple players. Due to the synchronization between the different download schedulers, different players will observe different bandwidths, leading to unfair shares of the available bandwidth. In order to address the synchronization issue, a randomized scheduler is proposed that relies on the buffer state (instead of solely time) as a basis for scheduling chunk downloads.

Stability in the playback session represents another issue in current players, which are prone to quality switches due to bandwidth variations. Given that stability affects the trade-off between efficiency and fairness, a delayed update method is proposed which may delay bit rate switches if there have already been a certain amount of switches in the recent history.

A more robust approach for bandwidth estimation is also suggested, which, besides considering averages across a set of previous samples, relies on the *harmonic mean*, instead of the traditional *arithmetic mean* which is often (mistakenly) used to compute average rates. This approach leads to a more robust average, less susceptible to outliers.

The extensive evaluation performed is able to clearly demonstrate that FESTIVE [244] outperforms the most common commercial implementation of HAS players, such as Microsoft's Smooth Streaming, Netflix's Smooth Streaming, and Adobe's players.

2.6.5 Challenges and opportunities in the optimization of HTTP Adaptive Streaming services

In addition to the analyzed optimization aspects on HAS technologies, there are other factors in HAS which must be considered, given their impact on the practicability, feasibility, and manageability of HAS-based multimedia delivery solutions.

Because HAS assumes that the same content is encoded in multiple representations and then split into chunks, both the number of objects being managed and the amount of storage space required grows with the number of desired representations. The growth in storage requirements is usually accompanied with increases in bandwidth costs, given that the content must usually be replicated throughout the CDN nodes. There is a markable increase of objects that must be accounted for and moved around the network.

As another side effect of the number of objects being managed, traditional HTTP caches, which are transparent to the content transversing them, have a hard time providing reasonable cache hit-ratios.

Consider the case of an HTTP cache which is used both for live and on-demand content. The live content will present a much higher popularity than the on-demand one, and will thus tend to stay in cache for longer. However, the actual relevance of live content decreases exponentially with time and should be removed from the cache after a short time period. This effect will, in practice, lead to caches that tend to stay fully occupied with mostly irrelevant content.

2.6.6 Conclusion

QoE is *the* key metric that any service or application exposed to end-users should prioritize. Taking this fact into consideration, this section provided an initial high-level definition of QoE according to reference sources and institutions in order to clearly differentiate it from the more common QoS perspective. A description of the two QoE assessment categories was provided, in order to clarify the difference between subjective and objective methods.

Given that the main focus of this Thesis is on OTT multimedia networks, with a particular emphasis on HAS video protocols, an in depth survey of QoE estimation methodologies is conducted, initially regarding the more general problem of estimating QoE on video content, and later on the particular estimation issues that arise on HAS QoE estimation, along with 3GPP's recommendations for QoE monitoring.

Having understood the existing issues and approaches to address QoE modeling and estimation, a thorough analysis is performed on how to optimize the QoE in the different end-to-end segments, namely, content preparation, content delivery, and content consumption, as all of these steps have an impact on the overall user experience.

Finally, an overview on the open research challenges is provided in order to illustrate issues that should be addressed by future research. From the key areas to improve, the issue of QoE evaluation on HAS protocols is the most important in the context of this Thesis' research work.

2.7 Data Mining

State-of-the-art multimedia services over the Internet, such as TV related offerings, are provided through complex large scale deployments to millions of daily users. These systems have a broad set of requirements, which from a user perspective may include Authentication, Authorization and Accounting (AAA), while from a service standpoint may encompass permanent availability and performance monitoring. All of these tasks, which may be perceived as secondary to the core service (e.g. Catch-up TV delivery), are essential to a production system and are characterized by at least one key common denominator: they generate vast amounts of logging data.

The process by which large amounts data is used to extract meaningful information, patterns, characteristics, and build models, is commonly denominated by *data-mining* [247]. These large chunks of data, or big-data, contain a multitude of useful information, and are suitable for several purposes, such as Intrusion Detection Systems (IDSs), Business Intelligence (BI), Business Analytics (BA), and subject areas that include social media, retail, finance, and telecommunications. From a network and service perspective, the data logs are suitable for optimization processes, which receive it as input for feedback loops and then act on the services' systems with the purpose of improving their performance or lowering their operating costs. Practical examples of how data-mining techniques may be used to improve services' performance are provided on sections 2.3.6, and 2.5.3.

The purpose of this section is to explore state-of-the-art approaches for handling vast amounts of data with the purpose of building data models that may be used to forecast services' demand. This predictive approach, which enables the anticipation of users' demand, is essential to perform proactive service optimizations and maintain the best possible users' QoE. An overview on the broad predictive data modeling process is conducted, followed by a detailed description of each step involved in the predictive modeling process, from data transformation up to model performance evaluation.

2.7.1 Predictive Data Modeling

Predictive data modeling is the area of *data analysis* that focuses on building models able to forecast *yet-to-be-seen* data, within a given prediction accuracy. This scientific research area is highly related to machine learning, pattern recognition and data mining fields, which are used to build the models.

The process by which a predictive data model is built follows the same industry guidelines as any other data mining process, i.e. the Cross Industry Standard Process for Data Mining (CRISP-DM) [248], as illustrated in Figure 2.24. CRISP-DM was defined by over 300 organizations. Its purpose is to encourage interoperability of data mining tools, and to simplify the data mining processes. The establishment of a reference process provides several advantages over custom tailored approaches, and include an easier replication of analyses, a lower barrier of entry for novice researchers, along with an encouragement for following industry best-practices.

There are seven main steps involved in generating and applying a predictive data

Figure 2.24: Main Steps Involved in Generating a Predictive Data Model.

---

model, which we be described in detail in the following subsections, nonetheless, the high-level goals of each individual step are as follows:

- *Data Acquisition*: The first and foremost step is that of data acquisition, which, depending on the amount, source and format of data may be challenging and costly;
- *Data Understanding & Exploration*: The next step is dedicated to understanding the data domain and the meaning of the data itself. In this step, it is desirable to establish data mining goals, to produce a project plan, to start describing the data available, to perform data exploration and to assess the overall data quality;
- *Data Pre-Processing*: Having established the data mining goals, the data must now be preprocessed in order cleanup redundant information, to construct new data that is more meaningful to the problem at hand, and to transform it so that it is better suited for processing.
- *Feature Selection*: With the relevant data already preprocessed, the next challenge is to perform feature selection, i.e. to determine which data fields provide the most relevant information towards the set goal, taking into consideration complexity trade-offs, as each additional feature tends to increase the computing requirements for model generation, and may not add much in terms of the overall solution performance;
- *Model Training*: The first challenge in model training is to decide which model

to use. There are hundreds of alternatives to be used in data mining, and more specifically on predictive data modeling. Popular models include SVMs, k-Nearest Neighbors (k-NNs), Neural Networks (NNets), and Random Forests (RFs); however, the choice of model is highly dependent on the characteristics of the data, on the desired goals, and on the type of prediction, i.e. *classification* or *regression*. Additionally, in order to avoid overfitting the model, special care must be taken in the training phase to increase its robustness when faced with unknown data;

- *Performance Evaluation*: This step is indissociable with model training one, as an iterative process usually exists between model training and tuning and the performance evaluation step. In order to perform a proper model evaluation, it is necessary to establish target goals with respect to desirable model *bias* and *variance*. *Bias* refers to how close the estimation is from the true value, while *variance* refers to how much the predictions differ from each other;
- *Deployment*: Lastly, the deployment step focuses on the challenge of using the prediction model and /or results to integrate them into a new or existing system.

A more detailed insight on these steps is provided in the ensuing subsections. The *Data Acquisition*, *Data Understanding & Exploration*, and *Deployment* stages are not considered, as they are problem-specific.

2.7.2 Data Preprocessing

Adequate data preprocessing is one of the key steps in building a predictive model that is able to accurately forecast outcomes for new unseen data, and can make or break its applicability.

Different machine learning algorithms have different sensitivity to predictors' characteristics; however, given that machine learning algorithms rely heavily on mathematics and numerical stability, it is important that their inputs are formatted to help them.

The data preprocessing stage is usually subdivided into three main phases, according to their purpose.

First, individual transformations on predictors are applied, in order to normalize their statistical properties, such as distribution curve, variance and mean.

Next, additional transformations are applied to sets of predictors with the purpose of dealing with outliers, binning predictors into categories or identifying highly-correlated predictors, which may not add new information to the problem at hand.

The last category is devoted to feature engineering, i.e. on methods for creating new predictors from existing ones that might improve the performance of learning algorithms.

Even though this section only addresses *unsupervised* data preprocessing techniques, due to their greater popularity and widespread usage, it is worth to notice that a complete research field on *supervised* approaches also exists. *Supervised* approaches use the outcome variable in preprocessing algorithms, while the *unsupervised* techniques do not.

Individual Transformation

Individual feature transformations act on each predictor to apply an individual mathematical transformation. Their purpose is to normalize the features so that they are numerically more similar to each other. This transformation helps machine learning algorithms by maintaining their numerical stability and by not favoring some predictors over other simply because of their different scales, for example. There are essentially two types of transformations: those that scale and center the content, and those that modify the distribution curve of the predictors, i.e. that perform skewness correction.

Scaling and Centering is one of the simplest transformation that may be applied to individual predictors. Scaling refers to adjusting the predictors so that their standard deviation is 1, thus dividing each value by the predictor's standard deviation. As for centering, the operation consists on subtracting each value by the predictor average, so the final predictor average value is 0.

Skewness Correction, on the other hand, addresses the issue of predictor with very different distribution curves, which also impact the performance of machine learning algorithms. An example of an un-skewed feature would be one that follows a symmetric distribution, such as bell curves. Traditional transformation include the application of square roots, logarithmic, or inverse functions. Figure 2.25 shows an example of two skewed distributions, along with a traditional bell distribution.

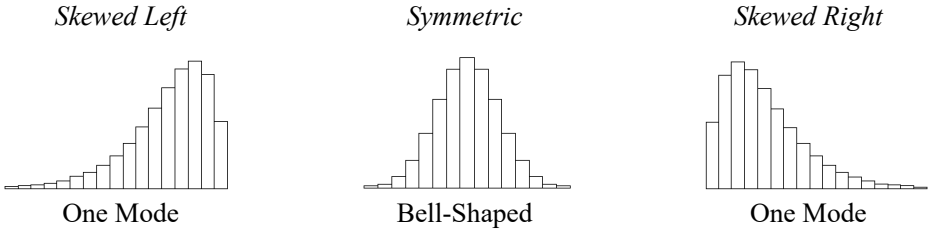


Figure 2.25: Example of Left and Right Skewed Distributions [6].

An indicator of skewness is the traditional *Fisher-Pearson coefficient of skewness* [6], presented on equation 2.2.

$$skewness = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^3}{\left[\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2\right]^{3/2}} \tag{2.2}$$

Alternative skewness correction methods exist that do not rely empirical analysis, but rather on statistical properties to identify the appropriate transformation. One of the most widely known and used transformation is the one proposed by Box and Cox [249], presented on equation 2.3, where parameter λ must be tuned to minimize skewness.

$$x^* = \begin{cases} \frac{x^\lambda - 1}{\lambda}, & \text{if } \lambda \neq 0. \\ log(x), & \text{if } \lambda = 0. \end{cases} \tag{2.3}$$

In spite of the popularity of the Box-Cox transformation, its lack of support for negative and zero values is a problem that lead Yeo and Johnson to propose modifications to the transformation so that its application could be expanded [250].

The resulting transformation might not be perfectly symmetric; however, it will have a much better distribution than the original data.

Multi-Predictor Transformation

In addition to processing individual predictors, the issue of transforming groups of predictors is also of great importance, and has the general purpose of identifying outliers, reducing data dimensionality, and dealing with missing values.

Outliers are defined as values that significantly differ from the rest of the data, and that may negatively affect the performance of predictive models. Depending on the data gathering process, or the underlying data generation system, outliers are to be expected in the source data of any data mining system. It is important to validate the data to ensure that their values have any scientific meaning, and that no errors occurred during the data acquisition stage. The decision to classify some values as outliers must be carefully considered, particularly in the case of small datasets, where the addition or removal of a few samples may significantly impact the performance of the predictive model. Before removing outliers, centering, scaling and skewness correction should be performed in order to reduce the chance of wrong removals. In order to mitigate the impact of outliers some mathematical transformations might be applied, such as *spacial sign* [251].

Missing Values may be present in some predictors of the original data, even if the dataset as whole is of good quality. The reasons for missing values are usually problem specific; however, because predictive models are mathematical models, most of them *need* their variables (there is the notable exception of tree-based techniques). Thus, entries with missing values must be either dismissed, or filled-in with valid data. It is necessary to understand if the absence of a value has any particular meaning – e.g. a measurement instrument that has a lower detection limit – and to address those specific situations by, for example, filling in the value with the inferior detection bound of the instrument. Depending on the dataset size, and on the representativity of samples with missing values, a good option might be to simply remove the sample from the data. If the missing values are in fact required, either due to the sample relevance, or to the size of the dataset, imputation methods exist for predictive models that are able to extrapolate them, as discussed in [252].

Data Reduction techniques focus on the problematic of data dimensionality, by combining predictors into a smaller subset of new ones able to convey the most significant information of the original data. A popular data reduction technique is Principal Component Analysis (PCA) [253], which tries to identify linear combinations of features that capture the most possible variance, and that are simultaneously uncorrelated with the previous ones. Each linear combination is denominated Principal Component (PC). By selecting only the PCs that account for a significant amount of the total variance (e.g. 99%), and use them as features, the overall dimensionality is reduced.

Predictor Binning consists on taking numerical features and transforming them into categorical ones, thus binning a given feature into two or more. This technique may have a direct negative impact on the performance of the predictive model, by reducing its precision and leading to a higher rate of misclassification [254]. In spite of these performance shortcomings, this transformation may be helpful to improve results interpretability, which in some applications might be worth the precision loss.

Feature Engineering

Feature engineering, or predictor addition, is a type of transformation that applies to both individual and multi-predictor transformations. It is a procedure through which existing predictors are encoded into new predictors conveying data differently. This approach has great applicability when the data at hand has features comprising categorical variables, such as age or gender. In the case, categorical variables are expanded into the so-called *dummy predictors*, which are numerical variables using group indicators of ones and zeros. A simple example is shown in Table 2.4 where a categorical feature comprising three age groups was encoded three new features. In practice, only two new features (instead of three) would be needed, as the third could be inferred.

Age	**Child**	**Adult**	**Elder**
Child	1	0	0
Adult	0	1	0
Elder	0	0	1

Table 2.4: Sample Dummy Variable Encoding.

In addition to being applicable to categorical predictors, feature engineering may also be useful to extract new features that may better model the problem at hand. A common example is that of date representation. Even though a date might be represented using, for example, a Unix timestamp, a predictive model might have a better performance if the timestamp is converted into separate day-of-week, day-of-month, and time-of-day predictors. Another example would be to represent the time difference between two dates as new predictors. Feature engineering tries; therefore, to extract more meaningful predictors from the existing ones, so that they are more directly applicable to the problem at hand. Naturally, performing adequate transformations requires a thorough understanding of the problem domain.

2.7.3 Feature Selection

Selecting a subset of the most relevant features out of a larger predictor set is essential in the predictive model development process. The challenge is to remove predictors that do not add a significant amount of information to the models, nor significantly impact their performance, thus being expendable.

Using features that do not provide a relevant contribution to the outcome result may hinder the performance of the predictive model from several perspectives. First, some predictive models are highly sensitive to uninformative predictors, which have a direct negative impact on their precision. Second, some models become non-tractable as the number of predictors used for training grows, hence the term *curse of dimensionality* coined by Bellman [255].

In addition to these issues, the storage, network, time, and computing overhead associated with managing and acquiring a large set of predictors represents a cost that should be avoided. Moreover, from a practical perspective, models relying on fewer predictors are often more interpretable, and statistically more attractive.

The problem of feature selection must, therefore, be addressed. A few popular and simple approaches have already been discussed in 2.7.2; however, several more exist that are able to perform a more sophisticated analysis on which features should be removed. Depending on whether the feature selection process depends on the model performance and/or outcome variable or not, two main categories of feature selection methodologies exist: *supervised* and *unsupervised* [256].

Unsupervised Feature Selection

A common unsupervised technique for removing predictors is to identify features that have *Zero Variance*. If the predictor only has a single value, it does not add any new information to the model, is completely redundant, and may be removed.

A *fuzzier* approach to predictor removal is based on *Near-Zero Variance*, i.e. instead of removing only the predictors that do not vary, try to also remove those that do not vary much. The problem with near-zero variance-based predictor removal, is to decide which predictors that do not vary much are important, and those who are not. This issue is solved by considering the frequency of unique values. If a predictor with near-zero variance only presents a small set of unique values, where the vast majority is the same, there is a high probability that the remaining unique values are not relevant. The general rule of thumb for removing predictors based on a near-zero variance is described in [257]:

- Less than 20% of unique values *and*;
- Ratio between the most frequent and the second most frequent must be superior to 20.

This heuristic must be validated through experimental validation on each case, but serves the purpose of providing an indication on which predictors should be considered for removal.

Finally, other common technique for feature removal is that of *Collinearity Analysis*, or *Between-Predictor Correlations*. This technique works by computing the cross-correlation of every feature and filtering those that present a high correlation, usually more than 95%. Given that highly correlated features behave similarly, it is speculated that they have redundant information, thus not contributing with new and unique information to the predictive model.

Supervised Feature Selection

This feature selection class is much broader than the unsupervised one that was just described, and, because this approach depends on the performance of the predictive model, it is also usually more complex and computationally demanding.

It is possible to further subdivide it into two main groups [258]: *wrapper* methods that focus on adding and removing predictors to find the combination that maximizes model performance, and may use genetic algorithms, simulated annealing, recursive feature elimination, and ensemble strategies [259] to name a few; and *filter* methods, which, conduct evaluations not dependent on the predictive models, and try to find relationships between the predictors and the outcomes in order to select an appropriate set of predictors [260]. Both approaches have advantages and drawbacks.

Filter methods are applied before training the model, hence they are usually less computationally demanding than wrapper methods. However, because they are influenced by the outcome variable instead of the predictive model performance they might not lead to a set of predictors that maximize the models' performance.

Wrapper methods focus on searching subsets of predictors in order to find the ones that maximize the models' performance, the difference between the wrapper methods is usually on the search algorithms utilized.

A common issue with both approaches is that of overfitting due to selection bias, in particular when the data set is small. In order to mitigate this risk, resampling techniques should be used and are the focus of the next section.

2.7.4 Resampling Techniques

Resampling algorithms provide a way of evaluating the model building performance using several alternate subsets of the data, and have the ultimate purpose of avoiding, or reducing, the risk of overfitting a predictive model. Overfitting a predictive model may be disastrous in the sense that an over-fit model will behave poorly when faced with new data other than that it was trained on. Because predictive models should be able to accommodate variations in the input features and still make accurate decisions, overfitting is an issue that must be considered when building a predictive model.

The high-level approach for resampling is quite simple. An iterative process selects different subsets of data for training and testing, and then computes the model's aggregate performance estimate for each combination of subsets. Although conceptually simple, in practice there are several resampling approaches that vary on how the different subsets are chosen [261, 262].

k-Fold Cross-Validation

k-Fold Cross-Validation splits the available dataset into k subsets (*folds*) of roughly the same size. Then, the model under consideration is trained using all datasets but one, which is then used in the subsequent prediction phase to gather performance metrics. This procedure is performed k times, and the gathered performance metrics are

summarized using statistical methods such as the mean and standard-deviation.

The special case where k equals the number of samples is called leave-one-out cross-validation (LOOCV), as only 1 sample is used in the prediction phase at a time.

It is common to repeat this procedure a number of times (5 to 10), each time generating k different subsets, and only then summarizing the performance results.

Research has shown that repeated k-fold cross-validation is an effective way of improving the precision of the trained model and reducing bias.

Monte Carlo Cross-Validation

This technique, also known as "leave-group-out cross-validation" is a variation of k-fold cross-validation, where the testing and training subsets are selected randomly before the training phase. Usually around 80% of the samples are considered for training and 20% for testing. Just like in k-fold cross-validation, it is common to repeat the test multiple times (more times than in k-fold cross-validation) in order to improve the precision of the results.

Bootstrap

The bootstrap process relies on sampling with replacement for creating the testing and training sets. The training set is created by repeatedly sampling the original dataset until the number of elements in the training set matches the number of elements in the original dataset; hence, the training set will likely have duplicated samples.

On the other hand, the testing dataset is created by selecting the samples that were not selected to the training phase, i.e. the *out-of-bag* samples.

By repeating the bootstrap procedure, an estimate on the predictive performance of the model may be achieved.

2.7.5 Classification and Regression Algorithms

As previously identified, there are essentially two types of predictive algorithms, according to their purpose: *classification* and *regression* models. *Classification* algorithms focus on assigning each sample into a discrete category or group, and are useful for identification purposes, for instance. On the other hand, *regression* models produce a continuous numeric outcome out of the input variables.

The ensuing analysis will focus on three popular and flexible predictive algorithms that may be used either for classification or regression problems, although hundreds of others exist [256].

Support Vector Machines (SVMs)

SVM is one of the most robust, flexible and powerful models. It was originally developed for classification in an industrial environment (AT&T Bell Laboratories) with a strong focus on real-world applications, and was later expanded to support regressive applications as well [263].

The underlying approach behind regression SVMs strives to find a function $f(x)$ that does not exceed the maximum deviation ε from the true value, while at the same time being as flat as possible, thus ensuring an error margin, also denominated error *tube*.

This approach assumes that such a function exists, which may not always be the case. To accommodate for error, it is possible to add a slack variable ξ to cope with infeasible constraints. The slack variable may be weighted by a cost parameter C to penalize large slacks. Figure 2.26 illustrates the ε tube and the ξ variable slack.

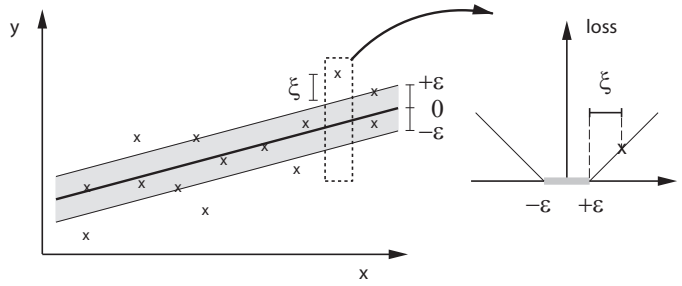


Figure 2.26: The soft margin loss setting for a linear SVM [7].

Even though the illustrated example demonstrates a 2D linear example, SVMs are able to generate regression and classification predictions on N-dimentional planes.

Computationally feasible support for N-dimentional prediction relies on the so-called *kernel-trick*, described in detail in [264]. Mapping the input space using a kernel leads to a new feature space, the hyperplane, where feature classification and regression becomes easier, as demonstrated in Figure 2.27. The choice of *kernel* to use in SVMs depends strongly on the problem; however, several general purpose kernels exist, including linear, radial, polynomial or Laplace.

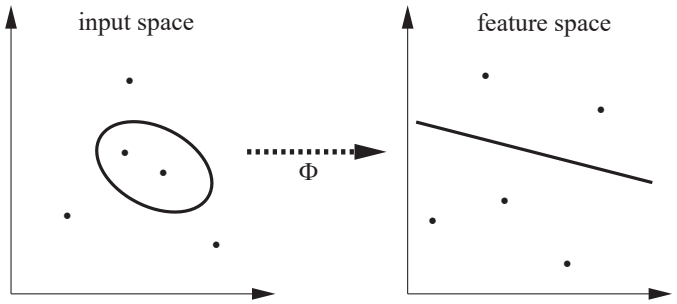


Figure 2.27: SVM - Input Space to Feature Space Mapping Using a Kernel [7].

k-Nearest Neighbors (k-NNs)

As the name indicates, k-NN predicts the outcome of new data using the k nearest samples used to train the model. The performance of k-NN depends on three important factors: determining k; selecting a distance metric; and choosing a value estimator.

Determining the k neighbors that maximize the model's performance can be performed through resampling techniques, where the value of k is iterated and, for each iteration, the Root Mean Squared Error (RMSE) is computed. The optimal number of neighbors would be the one that minimizes the RMSE. Small values for k tend to over-fit the model, while large k will under-fit it.

The distance metric plays an important role on the overall performance of k-NN, and is highly dependent on the problem domain. Common distance metrics include the Minkowski distance, described by equation 2.4, Hamming or Manhattan distances. The Euclidean distance is the special case of Minkowski distance where $q = 2$.

Minkowski's distance equation variables are as follows: q is the distance order between two points; P is the total number of features of a given sample; x_a and x_b are the samples whose distance is being computed.

$$\left(\sum_{j=1}^{P} |x_{aj} - x_{bj}|^q \right)^{\frac{1}{q}} \tag{2.4}$$

After determining the distance metric, and the k closest neighbors, the issue of selecting a value estimator depends on the nature of the predictor. Typically, either the median or the arithmetic mean are used; however, if the predictor represents a frequency, a more appropriate metric would be an harmonic mean.

Considering how k-NN works, it is important to ensure that the predictors are properly scaled and center, in order to prevent biasing the results towards predictors with larger scales. k-NN is easily interpretable and much simpler than SVMs, nonetheless, it may present computational time challenges and, depending on the training data quality, lackluster predictive abilities.

Neural Networks (NNets)

Neural Networks represent another widely used class of prediction models capable of generating predictions on non-linear data, while supporting sample classification and regression. In this brain-inspired technique, the prediction outcome is generated by weighting and combining the input data in a set of *perceptrons*, capable of applying linear combinations on the input data.

Figure 2.28 presents a classical multi-layer perceptron diagram, with four inputs, comprised of 3 layers: the *input* layer; the *hidden* layer; and the *output* layer. Even though the example only comprises one hidden layer, it is possible to have models with additional hidden layers, at the expense of computational time.

Even though the hidden units perform linear combinations on the input predictors, this combination is usually transformed using a non-linear *activation* function, thus enabling the support of nonlinear prediction models on NNets. Classical activation functions include the *Heaviside* - or step - function, the logistic sigmoid function, the normalized exponential function, and the hyperbolic tangent function.

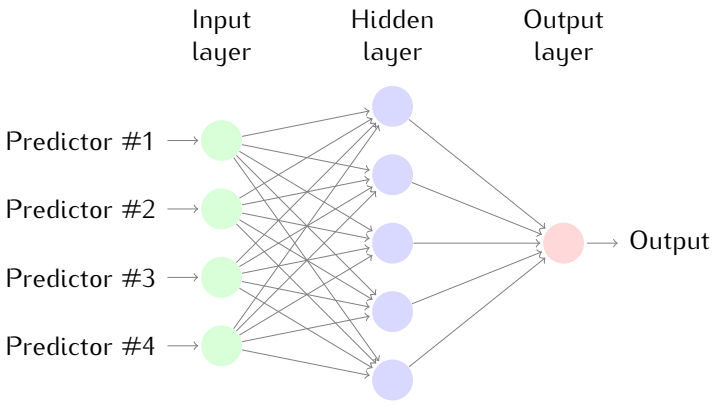


Figure 2.28: Sample Neural Network Multi-Layer Perceptron Diagram.

The inputs of the hidden units are then linearly combined in the output layer in order to generate a prediction outcome.

Just like in SVM and k-NN, the model optimization is performed by minimizing the RMSE, which requires solving equations initialized with random values through efficient solving algorithms, such as *back-propagation* [265] - which may not provide a global optimal solution.

Another side effect of neural networks is their tendency for overfitting the data, which may be mitigated by the *early stopping* technique that prematurely terminates the optimization process when the estimate of the error rate increases.

The performance of neural networks depends heavily on tuning efforts, including the selection of early stopping mechanisms, number of hidden units, and activation functions.

2.7.6 Performance Measurement in Regression Algorithms

In order to validate the effectiveness of the trained regression models and to compare them with others, it is necessary to be able accurately measure their performance, preferably from multiple perspectives.

One of the most common metrics for estimating the performance of a predictive regression algorithm is the RMSE, which provides an indicator of how far the prediction results are, on average, from the actual observed values, i.e. the prediction bias.

Another popular metric is R^2, which may be computed through several methods, but that is usually calculated by squaring the correlation coefficient of the predicted and observed values (R). This coefficient is good at providing an indicator on the data variation proportion that is explained by the model. R^2 does not, however, provide any measurement of accuracy.

A common approach for selecting a predictive regressive algorithm is to start with more flexible ones, such as SVMs, and then try find others that are more interpretable and still provide good enough results. *Good enough* might be defined by hard limits on the estimated prediction RMSE, for example.

2.7.7 Performance Measurement in Classification Algorithms

Given that the goal of classification algorithms is very different than that of regression ones, the previously used metrics, i.e. RMSE and R^2 are not suitable in this context.

Classification models generally output the probability of a given sample belonging to a particular class or group, that is then used to make a final classification decision. While production solutions tend to need the final discrete decision, in some applications it is more important to understand *how confident* the model is on a given class prediction.

Before taking any decision based on class probabilities, it is first necessary to ensure that the actual output of the predictive model has the same mathematical properties as a probability, i.e. that it is in the range of 0 to 1, and that the sum of all class predictions is 1. An example application would be on the output of NNets. In order to compensate for these issues, the *softmax* transformation is often applied [266], as described in equation 2.5, where p_l^* represents the transformed value between 0 and 1, y_l is the model's prediction for class l, and C is the number of possible classes.

$$p_l^* = \frac{e^{y_l}}{\sum_{l=1}^{C} e^{y_l}} \tag{2.5}$$

One important performance consideration is that the predictive model should produce class output probabilities that are comparable to the actual class probability, i.e. that the model is *well-calibrated*. This performance measurement technique may be performed analytically, or empirically using visualization tools, such as *calibration plots*, or *heat-maps*.

Another common solution is to use the so-called *confusion matrix*, which indicates the number of true positives, true negatives, false positives and false negatives. By gathering these statistics it is possible to analytically infer how the model is behaving with respect to its predicting accuracy.

There are, naturally, other performance measurement metrics that are domain-specific and depend on the application of the predictive model to real data in order to gauge its performance. An example would be the return on investment of a stock selection predictive model, or customer satisfaction improvements on proactive problem detection systems, to name a few.

2.7.8 Variance-Bias Trade-Off

When measuring performance on predictive algorithms, attention must be paid to the variance-bias trade-off (or dilemma), to avoid both over and underfitting.

The variance-bias trade-off translates an interpretation of under/overfitting, whereby the generalization error is decomposed into *bias* and *variance* [267].

In this context, *bias* refers to the tendency of a predictive algorithm to produce forecasts that diverge from the ground truth, while *variance* is associated with the predictive model's sensitivity to fluctuations of input parameters.

This trade-off is demonstrated by the dart-throwing example of Figure 2.29, which presents 4 possible bias-variance quadrants graphically. The bulls-eye represents the

ground truth, while each cross-mark depicts a prediction.

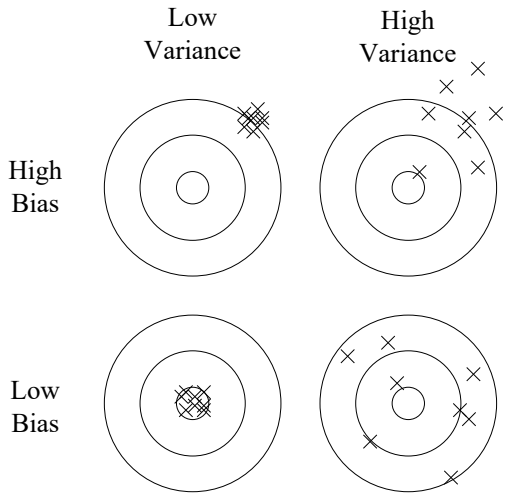


Figure 2.29: Example of the Variance-Bias Trade-off in Dart-Throwing [8].

Complex models tend to over-fit and usually produce predictions with higher variance than simpler models, but with reduced bias. On the other hand, simpler models will have a tendency for underfitting and not being able to accurately model the underlying system, leading to higher bias (i.e. less accurate), albeit with a lower variance.

2.7.9 Conclusion

The growth and ubiquity of IT has lead to an exponential increase in the volume of data generated by information systems. Data mining techniques capitalize on this data by identifying patterns and extracting meaningful information with the purpose of helping decision making. This discovery and modeling process has been the focus of several research initiatives over the past half century.

Properly understanding each data-mining step is critical to ensuring an adequate performance of the built models. Even though a common framework for data mining exists, i.e. CRISP-DM, most applications present unique requirements; thus, this section addressed the most important steps and methods involved in creating predictive models, from feature preprocessing, up to model performance evaluation.

Given the importance of dynamic, knowledge-driven, mechanisms, data-mining techniques present themselves as crucial tools to improve the performance of OTT multimedia delivery solutions in a proactive, rather than reactive, manner, and are considered in this Thesis' work.

2.8 Conclusion

This state-of-the-art review provides a thorough overview on the issues and technologies that have an impact on the QoE and performance of OTT multimedia services.

A description of existing OTT services is provided with a special focus on the services delivered by telecommunication operators on their Pay-TV offerings. This is essential to frame the requirements of an OTT delivery approach, which must be designed to meet the needs of the services being delivered.

To demonstrate the complexity of multimedia delivery solutions, a characterization of the content delivery pipeline is performed. By breaking down the complete system into parts with a clear separation of responsibilities, it is possible to gauge where improvements may be conducted by this Thesis' work, i.e. on the distribution macro block.

CDNs convey a great deal of responsibility in ensuring that relevant multimedia content is put close to users in order to maximize their service quality and address scalability concerns. They fit within the previously mentioned distribution macro block, and their architectures must be carefully designed to support next-generation multimedia delivery solutions in a scalable and efficient manner; thus, they are also a target for improvement in this Thesis which assesses the performance of different approaches, proposes optimizations, and evaluates dynamic provisioning features.

As multimedia content cannot be properly delivered without suitable streaming protocols, an overview of the existing and new protocols is provided, where it is shown that novel HAS algorithms are expected to keep growing in popularity. HAS protocols are radically different from traditional streaming mechanisms, and increase the strain on the underlying delivery systems by requiring additional storage, for the multiple representations, while increasing fragmentation due to breaking content into chunks.

QoE research in the scope of HAS protocols is still incipient, in spite of being a key concern of any service. Its estimation is shown be very different than on other traditional streaming technologies and is still an open-issue; therefore, QoE assessment for HAS protocols is a challenge that is also addressed by this Thesis.

Caches are a key component of CDNs which deserve special attention and are the focus of a dedicated section where a literature review shows that multimedia services are hard to tackle in an efficient and high-QoE approach, demanding additional research in the face of novel HAS streaming protocols.

A recurrent topic in modern IT systems is data-mining, which is used to leverage the vast amounts of data generated by OTT delivery systems to build knowledge that is useful to perform dynamic and informed optimization decisions on CDNs.

This chapter shows that a complete OTT multimedia delivery infrastructure optimization is not trivial and is full of open-research challenges that this Thesis addresses to enable a next-generation delivery architecture capable of ensuring the performance, scalability and cost-effectiveness that future services will require.

Chapter 3

Characterization of Catch-up TV Services

The optimization of multimedia services is highly dependent on the characteristics of each individual service. A very popular class of multimedia services is that of Catch-up TV, whose popularity is on the rise within Pay-TV services. Their expressiveness and widespread usage make them a perfect candidate as a use-case for content delivery optimization on OTT CDNs.

This Chapter provides three key contributions towards this Thesis' research goals. First, the paper *Time-shift services: a taxonomy and techno-business impacts of Catch-up TV* [25] evaluates the technical and business relevance of Catch-up TV.

Next, the study *Survey of Catch-up TV and Other Time-Shift Services: A Comprehensive Analysis and Taxonomy of Linear and Nonlinear Television* [27] digs deeper in a worldwide survey which proves that this class of services has a high worldwide penetration, and that its relevance is expected to keep growing in the coming years.

Finally, the article *Catch-up TV Analytics: Statistical Characterization and Consumption Patterns Identification on a Production Service* [28] provides a thorough characterization of content demand patterns in a popular Catch-up TV service, with the purpose of extracting key insights that might be used to optimize the delivery of these multimedia services in an OTT context.

3.1 Time-shift services: a taxonomy and techno-business impacts of Catch-up TV

This section addresses a literature gap by providing a detailed evaluation on why Catch-up TV services are relevant to consumers, Pay-TV and content providers. In doing so, their importance is highlighted, and a strong case is made regarding their role in the television industry, which is quickly converging to OTT-based delivery.

Detailed information may be found in *Appendix A*, containing the full publication.

3.1.1 Introduction & Motivation

Television is undergoing a rapid process of changes and transitions [268]. From the audience point of view there are several factors that are changing the way television and other videos are consumed: new larger and thinner screens; multiple devices able to receive signals from broadcast and on-demand; the potential for sharing recorded programs between those devices; the IoT, which connects all digital devices in the home and on the road; and the audience, which used to be collective and concentrated in the living room, that now happens anywhere, anytime and using any device.

These changes are accelerated by Pay-TV services, which have been established as primary sources of access to new television technologies, even in emerging markets. The cord-cutters phenomenon, where people give up their Pay-TV subscriptions replacing them by OTT services [269, 270], requires quick reactions of all links in the television production chain [271]. Thus, the whole TV market is influenced by the technologies used in the Pay-TV offerings, including free-to-air stations.

A significant consequence, noticeable in this scenario, is a change in the way people watch television. Nowadays, clients of advanced Pay-TV systems have multiple and straightforward ways to watch time-shifted TV content, blurring the line between the consumption of linear TV and deferred (previously aired) TV content. More and more TV operators are offering worldwide manual and automatic recording features, with the local storage capacity being moved to the cloud.

The potential techno-business impacts of the Catch-up TV service are addressed in this work, along with the market motivations that Pay-TV providers should take in consideration. In addition, a description is presented regarding the potential considerations of content providers.

3.1.2 Scientific Contributions

This work addresses the previously described scientific research gaps by providing the following contributions:

- Clear definition and differentiation of time-shift services terminology and use-cases;
- Impact analysis of Catch-up TV service offerings from multiple stakeholders' perspectives: users, service providers, content providers and linear TV services.

3.1.3 Taxonomy of Time-shift TV services

Time-shift TV relates to the visualization of deferred TV content, i.e. linear-TV content that is recorded to be watched later – from seconds up to several days –, using one of the following services:

1. *Pause TV* is the simplest type of time-shift service, allowing users to pause the television program they are currently watching - from a few seconds to several minutes or even hours. Users can resume the TV broadcast when they want, continuing to watch where they left off; skip a particular segment; or eventually catch up to the linear broadcast.
2. *Start-over TV* enables users to restart programs that have already started and, eventually, programs that already finished. The amount of time that is possible to rewind varies from operator to operator ranging from some minutes up to 36 hours. The number of TV channels supporting this feature is also a decision of the TV operator.
3. *PVR* stands for Personal Video Recorder. In this type of service the recordings are subject to the user action, i.e., they only occur if the user proactively schedules a TV program or a series to be recorded, or if he decides to start recording a program that is being watched. The behavior of the service is much the same as the one of a VCR (Video Cassette Recorder); however, with a much higher storage capacity and nonlinear access. The user can start watching a recording whenever he wants, even if the program is still being recorded.
4. *Catch-up TV* is the most advanced time-shift service, relying on an automated process of "Live to VoD" [39] (offered by companies like Alcatel-Lucent [40]) or on a more restricted process-based editorial control. With this service, TV operators offer recorded content of the previous days, on a bouquet up to hundreds of TV channels. The time window of the recordings ranges from a couple of hours up to 30 days, and the number of recorded TV channels varies from operator to operator, according to technical, legal, and business constraints. Using this service, users can really, and very easily, catch-up TV programs that have been missed or that they explicitly decided to watch later – e.g. watching the news after preparing dinner.

3.1.4 Why should Catch-up TV be offered to Pay-TV customers?

Catch-up TV is the reflex of content-centric paradigms where the content, and not the TV station or the airing time, is paramount. Because Pay-TV industry is supported on complex relationships between multiple stakeholders, as may be observed in Figure 3.1, the decision of adding a new service must be carefully analyzed in order to consider the established balance of power, and to assess its impact along the complete supply chain, where each stakeholder is affected differently.

The main business value proposition of Catch-up TV services lies in consumer empowerment. The control of what to watch, and when, is transferred from the broadcasters to the consumers, disrupting the established editorial control, and increasing consumer choice. In a time where cord-cutters [270, 269] are a reality, paying attention to customers

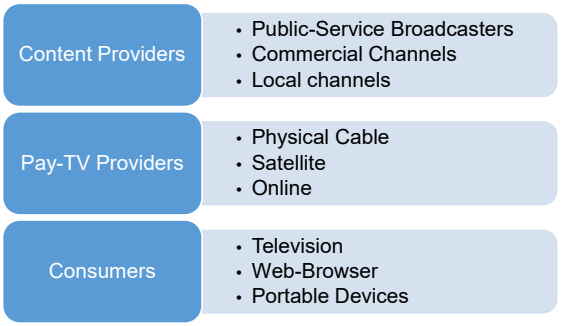


Figure 3.1: Pay-TV Industry Supply Chain.

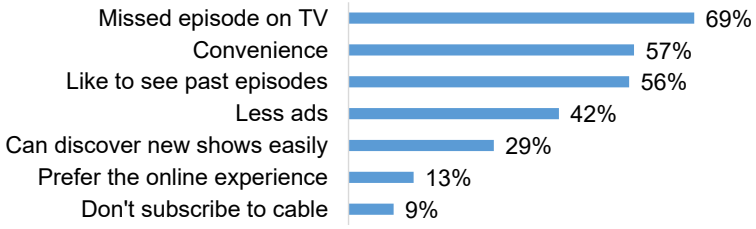


Figure 3.2: Reasons to Watch Video Online [9].

is crucial to improve their satisfaction with Pay-TV services, hence fostering customer acquisition, retention, and upselling. For Pay-TV providers, preventing cord-cutting, reducing churn, and increasing the Average Revenue Per User (ARPU) is essential and requires a rich and convenient service offering. A positive impact on ARPU caused by Catch-up TV has been shown in [272].

The work in [273] shows that most consumers have been clients of their Pay-TV service provider for less than 5 years, which indicates that the market is highly dynamic and that users are willing to switch providers in order to take advantage of added features, improved user experience, higher content quality, and lower prices. For example, Belgian operator Proximus' annualized churn rate on triple-play services was 10.5% on its first 2015 quarter [274], up from 9.3% on the previous quarter [275].

To determine what features present an appealing value proposition, a possible approach is to look into the reasons that drive consumers out of the Pay-TV experience into alternative media services, such as online video. ComScore data [9], displayed on Figure 3.2, indicates that the main reasons for watching online content are missed TV episodes and the desire to watch past episodes of TV shows.

Broadcasters also benefit from user engagement in Pay-TV services, as the amount of advertisement watched by users and its cost is much higher than on other comparable services, as is clearly visible in Figure 3.3 and Figure 3.4.

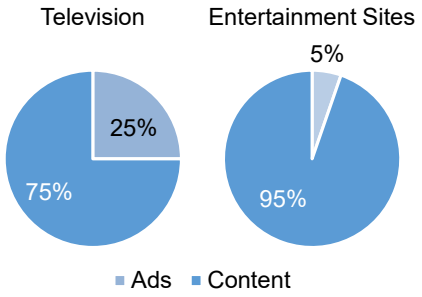


Figure 3.3: Percentage of Time Spent Watching Ads [10].

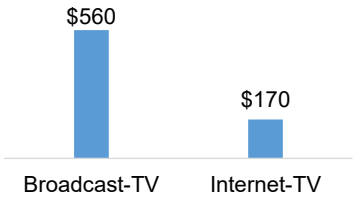


Figure 3.4: The Value of Broadcast vs. Online Viewers [9]. Advertising Value per Thousand Viewers per Episode.

Impact on Pay-TV Service Providers

While the benefits to consumers are well established, Catch-up TV has a significant impact on service providers' operations, presenting challenges of technical, economic, and legal nature. This is a service for the masses [42] with a high impact on the distribution infrastructures, as traditional broadcasting methods, using multicast on IPTV networks, do not work. The need to unicast video streams imposes severe network capacity requirements, which must be addressed by large investments [276]. The fact that Catch-up TV is data-intensive is also challenging, mostly because users are often not charged for the amount of data that needs to be transmitted in the network.

In addition to technical challenges, there are also licensing issues, as content providers may impose restrictions on the content available on Catch-up TV and require additional fees. Depending on each country's legislation, and on existing agreements, adding TV channels to the Catch-up TV lineup may be challenging [277].

In countries where Pay-TV providers offer a wide range of channels in Catch-up TV, they do so on the premise that it is a kind of Network Personal Video Recorder (NPVR) service, where the customer schedules full-channel recordings instead of just some shows. This seems valid for countries where NPVR shared copy is allowed and no additional compensation is due to the content owners. On the other hand, in many countries in America and Asia, where law mandates private copy [278, 279], this full lineup for Catch-up TV services does not exist.

Impact on Content Providers

Content providers decide the content price, thus having a high bargaining power in the supply chain, which is used to leverage new delivery forms as an opportunity for increasing revenue, such as demanding micro-payments (Pay-Per-View (PPV)), or charging additional fees per delivery service. However, limiting the availability of Catch-up TV content may be counter-productive. The reasons vary depending on the TV stations' business models.

For premium TV stations, where the advertising revenue is residual and most revenue comes from user subscriptions, not allowing a service like Catch-up TV reduces its value proposition, especially if the aired content does not have any temporal relevance, which is usually the case of movies and series premium channels, but also applies to sports channels, or other TV stations where live events are particularly important.

Regarding non-premium TV stations, whose main stream of revenue originates from advertisement, the Catch-up TV proposition is also relevant. Several studies show that, in spite of a reduction in linear TV viewing, in favor of time-shifted viewing, the overall television consumption has increased [272, 38, 280] due to time-shifting services. [281] shows that if Catch-up TV were a TV station, it would be the most popular on prime-time.

Non-premium TV stations fear that the reduction on linear TV consumption will lead to a reduction on advertisement value, thus having a negative impact on revenue. However, it has been shown that not all users skip advertisements, and that advertisements get up to 44% more views due to time-shifted viewing [282]. Additionally, the vast majority of advertisements are still relevant on most Catch-up TV reproductions, which happen mainly within 3 days of the original airing, regardless of the total Catch-up TV window [272, 283].

Ultimately, because Catch-up TV increases media consumption, content providers get an increased exposure of their programs, and advertisements, to consumers. This motivated Nielsen [284] to release the so-called "C3" ratings that encompasses commercials watched both live and in a 3 days window, which show that some content, like serialized TV shows, get boosts of more than a full rating point. More recently, new metrics increased the commercials' analysis time window up to 35 days [285].

Impact on Linear-TV

One of the myths regarding Catch-up TV services is that they significantly reduce the consumption of linear television. While it has been shown that users watch less linear television in favor of other media, the difference is not significant (-2% over a two years period), and linear TV continues to be as relevant as before [38].

Even though this reduction occurs, the programs are still watched. The most popular programs in Catch-Up TV are the most watched in linear TV. [272] found that prime-time content is the most watched content during prime-time and off-peak hours on nonlinear TV. This finding suggests an increased overall viewership of prime-time content in detriment of other content.

The work in [286] claims that Catch-up TV is a natural consequence of television evolution. With the digitization of the production, transmission and reception, the value chain becomes flexible, allowing new features and services offerings. Thus, two consumption scenarios arise: time-shift services address content without significant temporal relevance; and linear TV focuses on programs with immediacy appeal.

3.1.5 Conclusion

Catch-up TV is the most advanced time-shift technology, presenting a remarkable potential for changing viewers' relation with TV.

From a business perspective, preventing the cord-cutting phenomenon, reducing churn, and increasing the ARPU, is essential and can only be achieved by providing a rich and attractive service offering empowered by Catch-up TV services.

From a technological point of view, broadcasters could offer all programs simultaneously in the cloud, and the viewer could choose what and when to watch, regardless of the transmission time, from a much larger TV content offer. That is, content and service quality becomes the differentiation factor, and not the lack of choice or program transmission time.

Therefore, given the importance of removing technological limitations to provide flawless Catch-up TV services, particularly in OTT scenarios, where it is harder to provide quality guarantees, it is essential to explore potential delivery optimization opportunities.

3.2 Survey of Catch-up TV and Other Time-Shift Services: A Comprehensive Analysis and Taxonomy of Linear and Nonlinear Television

As a follow-up to the previous research work, this article, available in *Appendix B*, provides a worldwide survey on Catch-up TV and other nonlinear services, while simultaneously framing their differences and delivery technologies.

As is the case of the previous work, it reinforces the importance of nonlinear TV services, and establishes a strong understanding on services that are expected to be a part of a next-generation OTT marketplace.

3.2.1 Introduction & Motivation

Technology plays a crucial role in the television usage and value generation in the broadcasting market [287], thus, a great deal of attention has been given to the development and introduction of new technologies [49, 45], to changes in audience behavior [51, 288], and to impacts on market and business models [289, 290]. Studies have been conducted on how content recording impacts different countries [291, 289], however, even though some studies reported an increased usage of Catch-up TV, there has been little research on the international market and scientific community on how to organize and classify these new services. Television analysis is usually focused on local research, with limited implications and conclusions regarding international offerings of Catch-up TV, VoD and Over-The-Top (OTT) services [50, 42].

This study examines Catch-up TV and other nonlinear services in 62 countries, spread across 4 continents, to identify and quantify their availability on Managed Operator Networks (MONs), and shows that the nonlinear services are becoming ubiquitous.

When offered through the TV set, Catch-up TV provides a significant contribution to a great user experience. This unique characteristic paves the way for a remarkable worldwide penetration, as demonstrated by the fact that the first commercial releases have no more than 9 years, and already represent a first class feature in a very significant number of countries – 74 operators from 34 countries provide it.

3.2.2 Scientific Contributions

The key scientific contributions provided in this research work are summarized in the following list:

- Proposal of a detailed taxonomy of ways of watching TV, considering linear and nonlinear content delivered through managed or OTT networks;
- Survey of nonlinear time-shift services in 62 countries, spread across 4 continents, to identify and quantify their availability on MONs;
- Insights on cost, performance, and technological aspects of time-shift TV services.

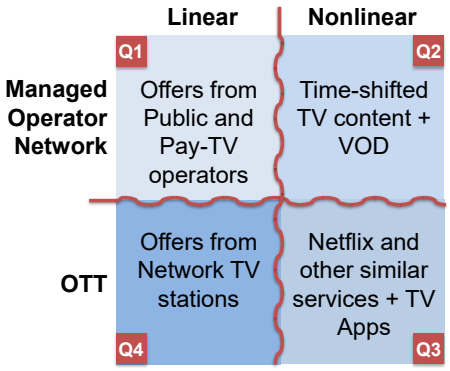


Figure 3.5: Four Major Quadrants of Ways of Watching TV.

3.2.3 A Taxonomy of Ways of Watching TV Content Over the TV Set

The frontier between linear TV and other forms of watching TV content is blurring, as is the corresponding terminology, which is becoming less clear and consistent not only among the different players of the TV ecosystem but also within academics. In order to provide a clear understanding of the ways of watching TV content on the big screen, Figure 3.5 depicts a matrix with 4 quadrants. The columns separate linear and nonlinear content, while the rows distinguish managed and unmanaged OTT network delivery.

This macro organization is not completely hermetic, since some services that are put into a particular quadrant might be found on neighbor sectors, although with a minor relevance. Regarding transmission types, the focus is on legal broadcast and streaming offers, and do not contemplate download and play content.

Linear Content over Managed Operator Networks (Q1)

Linear TV, i.e. "regular TV broadcast" respecting a predetermined program lineup [292], was considered for decades as the traditional and more popular way of watching TV programs. This is still the dominant way of watching TV from national free-to-air TV services and major Pay-TV Operators; however, customers are moving to other quadrants as detailed in the forthcoming sections.

Nonlinear Content over Managed Operator Networks (Q2)

With the advent of interactive services supporting deferred TV content, e.g. Catch-up TV, major Pay-TV operators started offering time-shifted TV in addition to VoD content through user-friendly TV interfaces. Unlike the straightforward classification of services belonging to Q1, the time-shifted TV and VoD categories are highly dependent on their business and technology characteristics, as presented in detail on Section 2.2.2.

Nonlinear Content via OTT (Q3)

Q3 shifts from nonlinear TV content offered by Pay-TV operators to content (mostly movies and series) delivered over the Internet without the involvement of Pay-TV operators. Bridging devices, such as computers, smartphones or tablets, are central to this scenario, and to some extent to the linear scenario of quadrant Q4 as well.

Linear Content via OTT (Q4)

In this quadrant, there are different approaches for watching linear-TV over the Internet, i.e. in an OTT way. Traditional broadcasters and Pay-TV operators tend to offer web sites, dedicated applications and players, while recent competitors provide pure-OTT alternatives such as Sling TV [293], or PlayStation Vue [294]. In the latter case, these offers are independent from any Pay-TV operator, their customers are real cord-cutters, relying only on an ISP contract for watching linear-TV for a free or small monthly fee. TVPlayer [295] is an example.

3.2.4 Worldwide Overview of Services Offering Nonlinear TV Content over Managed Operator Networks

Considering this work's focus on the exploration of new viewing practices of nonlinear TV over MONs, a survey is performed on the worldwide offer of services belonging to quadrant Q2 of the taxonomy proposed in the previous section – Figure 3.5.

The dominant potential of Catch-up TV services and their impact on the TV ecosystem is presented in this quadrant. Catch-up TV has a strong impact on the TV ecosystem, and significantly contributes to a great user experience, as demonstrated by its worldwide penetration growth since 2007 [296, 297].

Data gathering methodology

A systematic methodology is employed to perform a thorough overview of Pay-TV operators in Europe, America, Asia and Oceania supporting nonlinear TV services. Using a worldwide list of Pay-TV operators, the web-sites of major providers from 62 countries are visited. When applicable, Google's automatic translation tool is used.

Due to interest on the current footprint of Catch-up TV services, operators offering the service are listed in a spreadsheet with the following key fields: country; operator; Catch-up TV product name; and time window of previously aired programs.

As for Catch-up TV details, following time windows are considered: up to 3 days; between 3 and 7 days; more than 7 days; and "other" when the time span depends on independent broadcaster agreements. In addition to Catch-up TV data, the spreadsheet available in [298] includes other time-shift TV services provided by the operators at stake: Pause-TV, Start-over TV and PVR supported on the local HDD (DVR) and on cloud based storage (NPVR). The availability of T-VoD, EST-VoD, and S-VoD services is also reported.

Results

As shown in Table 3.1, from the 62 countries analyzed, 34 have one or more operators offering Catch-up TV services, whereas in the remaining 28 countries they are absent, to best of authors' knowledge. Most operators with Catch-Up TV services also offer Pause-TV, Start-over TV, DVR and T-VoD, while NPVR, EST-VoD and S-VoD are less widespread – the characterization of these different nonlinear services is presented in Section 2.2.2. The infographic of Figure 3.6 was produced using the survey data.

Europe In Europe, from the 30 countries analyzed, 20 already offer Catch-up TV services, while 37 major Pay-TV operators offer Catch-up TV, with a prominence in England and Portugal. With respect to countries where Catch-up TV services are not found, there are some where legal issues are an obstacle.

These 37 operators also support Pause-TV, while Start-over TV and DVR features are widely available. Only 6 of the considered operators offer cloud-based PVR, i.e. NPVR. VoD services are not offered by 3 operators with Catch-up TV, which may be justified by their business models and expected users' adoption. Transaction and Subscription VoD (in most cases based on an integrated Netflix offer) are the most common forms of VoD, whereas Electronic Sell Through VoD is only offered by 3 operators.

Asia From the 15 Asian countries analyzed, a total of 12 operators are identified, offering Catch-up TV services in countries like India, Japan, Indonesia, Malaysia, Singapore and South Korea. A significant part (50%) of Asian Catch-up TV services are integrated in a special section of the VoD catalog. Every operator offers T-VoD and Pause-TV features, while Start-over TV and DVR are offered by most but not all.

America 15 countries and 23 operators are analyzed in the Americas. In most operators (83%), Catch-up TV services comprise a selection of channels/programs integrated as a special section of the VoD catalog.

Two differences stand out when compared to Europe. First, the amount of available programs is smaller, as in a regular Catch-up TV service in Europe users may access most programs of the subscribed TV channels. Furthermore, in most European Catch-up TV offers provide a dedicated user interface easing the retrieval of aired programs.

The Catch-up TV time window is dependent on agreements with each broadcaster.

As for the remaining time-shift TV services, all the 23 considered operators provide Pause-TV; Start-over TV and DVR. Only 3 provide a Network Personal Video Recorder. All but 1 operator offer VoD services, and the predominant type is T-VoD.

Oceania In Oceania, the survey focuses on Australia and New Zealand. In these countries, two operators are found that offer a Catch-up TV service with a time window of 7 or more days.

In addition to providing Catch-up TV services, these operators also support Pause-TV, DVR and T-VoD, although none provides Network Personal Video Recorder.

	# Countries analyzed			# Operators with Catch-up TV and other nonlinear TV services							
	Total	with Catch-up TV	without Catch-up TV	Catch-up	Pause	StartOver	NPVR	DVR	T-VoD	EST-VoD	S-VoD
Europe	30	19	11	37	37	34	6	29	34	3	15
America	15	7	8	23	23	23	3	23	20	0	10
Asia	15	6	9	12	12	7	1	8	12	0	1
Oceania	2	2	0	2	2	0	0	2	2	0	0
Total	**62**	**34**	**28**	**74**	**74**	**64**	**10**	**62**	**68**	**3**	**26**

Table 3.1: Survey of Catch-up TV and other time-shift TV services.

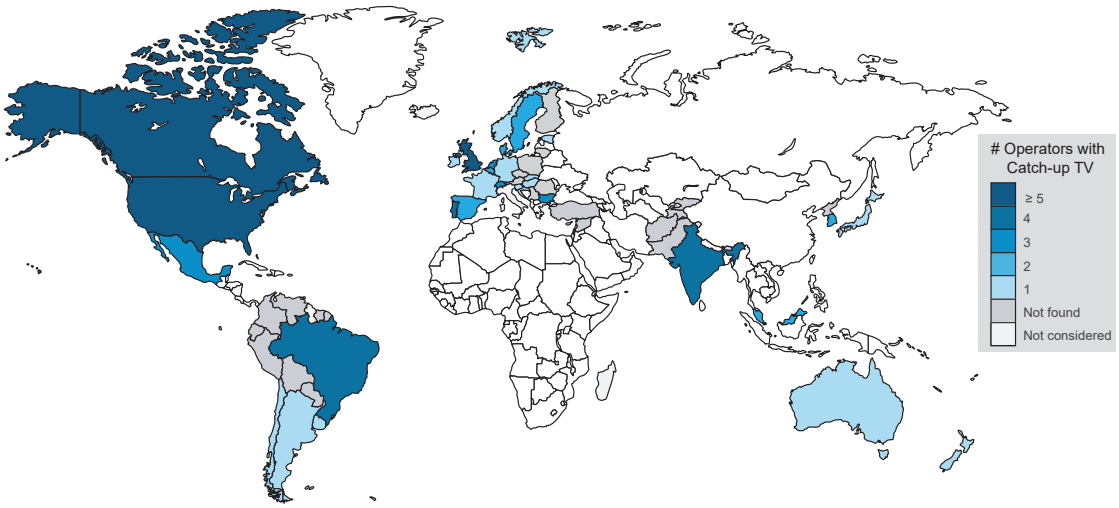


Figure 3.6: Overview of operators offering Catch-up TV and other time-shift TV services.

3.2.5 Conclusion

This thorough overview made clear that the current worldwide footprint of the Catch-up TV service is very expressive. The basic technology is widely available and users' adoption shows that this is a trend with the potential to spread into other countries and operators. Another aspect that stands out is the fact that other time-shift TV services (Pause-TV, Start-over TV and PVR) as well as VoD services are a constant in the offers of Pay-TV operators.

The presence of all these services over MONs is proof that users value the possibility of consuming TV content at their pace in a nonlinear way, especially if they have the opportunity to easily enjoy a service like Catch-up TV, which automatically records the content they want.

This study demonstrates the expressiveness of Catch-up TV and its importance in the Pay-TV ecosystem. Considering its current and growing relevance, Catch-up TV presents itself as a suitable use-case for exploring optimization opportunities in OTT multimedia delivery solutions, which motivates the ensuing research works.

3.3 Catch-up TV Analytics: Statistical Characterization and Consumption Patterns Identification on a Production Service

This study, presented in full in *Appendix C*, provides an in-depth analysis on Catch-up TV content consumption by leveraging a large dataset from a commercial service.

The results improve the current scientific knowledge on Catch-up TV, by characterizing users' behaviors and preferences, with the purpose of improving OTT services.

3.3.1 Introduction & Motivation

To keep-up with a growing demand, IPTV operators are turning to OTT delivery solutions which do not require investments on managed IPTV infrastructure, and increase the reach of services that may have been previously limited to certain geographic areas.

However, this move requires overcoming several challenges – as shown in Chapter 2. Given the different requirements of OTT delivery, when compared to that of managed networks, a thorough understanding of service usage is required to properly decide on OTT CDN architectures, plan the physical and logical location of clusters and replica servers, tune caching algorithms, select optimal request routing mechanisms, and estimate computational, network and storage requirements, to name a few.

In addition to OTT-specific service improvements derived from utilization data, the characterization of Catch-up TV consumption presents several optimization opportunities, both from users' and operators' standpoints, regardless of the delivery approach. An optimized service improves users' QoE and overall service satisfaction, which is essential to prevent churn in modern and highly competitive Pay-TV markets.

A thorough understanding of demand patterns fosters operators' savings on CAPEX and OPEX. CAPEX is reduced by investing on less extra capacity, because the exact service requirements are known and the delivery system is optimized to meet them, which also contributes to reducing the OPEX. Other potential savings come in the form of energy efficiency, achievable by using elastic resources taking advantage of content consumption patterns to provision only the required resources, or even by loading content in advance into users devices with the purpose of lowering peak resource demand.

In summary, an exhaustive modeling of Catch-up TV content consumption patterns enables a great deal of optimization opportunities, and is thus the focus of this work.

3.3.2 Scientific Contributions

- Detailed statistical study of a large scale Catch-up TV consumption dataset with 21 different analyses on programs, users, and behavioral characteristics;
- Assessment of content delivery optimization opportunities from caching and bandwidth requirements perspectives;
- Presentation of detailed summary tables that may be used by the scientific community to create statistical models for Catch-up TV consumption.

3.3.3 Dataset Description

A Catch-up TV consumption dataset is collected from a major IPTV operator and contains 30 days of program request logs, regarding the full month of April 2015.

This nonlinear service provides free access to the previous 7 days of program airings on 80 TV channels, depending on users's subscriptions. The content is delivered through a managed network infrastructure using RTSP streams. Even though it would be desirable to have information on users' genre and age, the fact that the TV is commonly shared by several family members, and that the IPTV service in question does not support user profiles, prevents a targeted analysis.

By combining the data with the Electronic Programming Guide (EPG), each request log entry enables a rich characterization of an individual playback session. Any information that might reveal user details is anonymized. Time and date fields are in Greenwich Mean Time (GMT) timezone. Each item has the following form:

- *Account Id* - Unique user account identification;
- *Set-Top-Box (STB) Id* - Unique STB identification to distinguish requests from different devices in the same household;
- *District* - Geographical information containing the household location (district);
- *Title* - Name of the requested program;
- *Station Id* - Unique TV station identification on which the program aired;
- *Station Genre* - Falls under the following categories: *General*, *Sports*, *Kids*, *Documentaries*, *News*, *Movies And Series*, and *Entertainment*;
- *Station Video Quality* - Video quality indication of the TV channel: either High-Definition (HD) or Standard Definition (SD);
- *Program Id* - Unique program identification within the EPG;
- *Series Id* - Unique TV series identification within the EPG;
- *Season Number* - If the program is a TV Series, its season number;
- *Episode Number* - If the program is a TV Series, its episode number;
- *Start Time* - Original broadcasting start time of the requested program;
- *End Time* - Original broadcasting end time of the requested program;
- *Play Time* - Timestamp of a playback session start.

These data fields are sufficient to extrapolate additional information, such as the playback day of week and content duration, for example.

Data Cleaning

Considering that the raw data is generated from systems that may be unreliable, produce duplicate entries, and contain records from test accounts, an initial data cleaning process is performed to remove data that does not accurately reflect the service usage:

- Removal of data originated from test accounts;
- Removal of duplicate entries;
- Dates and times are adjusted to the Portuguese mainland timezone.

After performing these data cleaning procedures, the key data indicators for the available dataset were extracted and are presented in Figure 3.7.

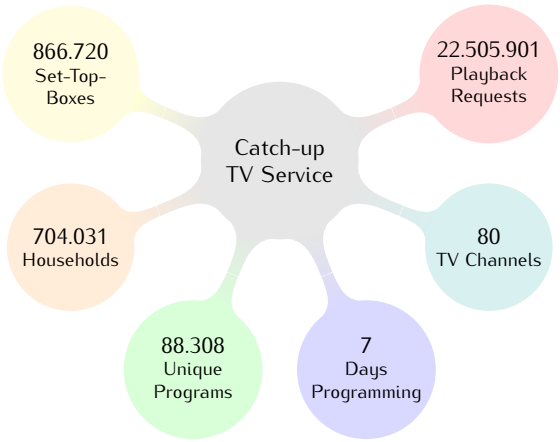


Figure 3.7: Catch-up TV Dataset: Key Data Indicators.

3.3.4 Main Results

This section presents the key findings of this study. The data presented in the figures is normalized so that 100% represents the maximum value, and 0% the minimum value. This normalization maintains the proportionality relationship between the multiple values and does not affect a critical analysis, but avoids disclosing absolute numbers. Additional analyses may be found in the expanded work contained in *Appendix C*.

Usage By Hour of Day

Figure 3.8 presents the number of program requests per day of week and hour of day, to foster a better comprehension on *when* users request programs.

Starting with a global examination on the characteristics of each individual plot, it is possible to conclude that users are less active on the late night-hours, approximately from 02:00 to 07:00, and use the service more intensively from 08:00, up to a peak at around 21:00, regardless of the day of week. The 02:00 to 07:00 interval corresponds to the normal sleeping hours, while the 20:00 to 23:00 interval matches prime-time.

On regular weekdays, the service utilization shows a continuous growth from 08:00 to the prime-time, while on weekends the service utilization is roughly constant throughout the day, with the exception of late night hours. This is as expected, as on weekends users are at home and watch Catch-up TV throughout the day.

Original Airing Time Relevance

This analysis is complementary to the previous one, in the sense that instead of concentrating on program request times, the key metric is the original broadcasting time, i.e, the day of week and time of day when the Catch-up TV program originally aired. Figure 3.9 shows that content aired on prime-time is also the most popular Catch-up TV content, exceeding the popularity of content aired on other hours of day. Additionally, the results also show that prime-time content of Fridays, Saturdays and Sundays is more popular than prime-time content aired on other days.

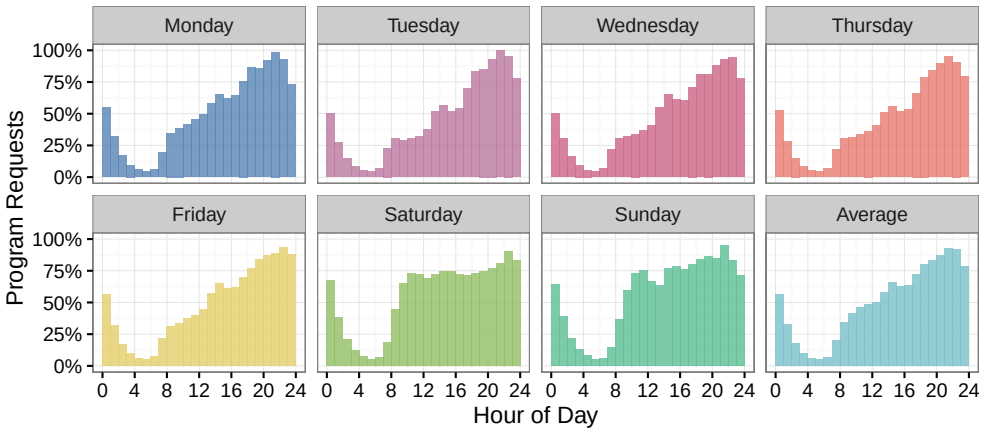


Figure 3.8: Service Usage: Day of Week and Hour of Day.

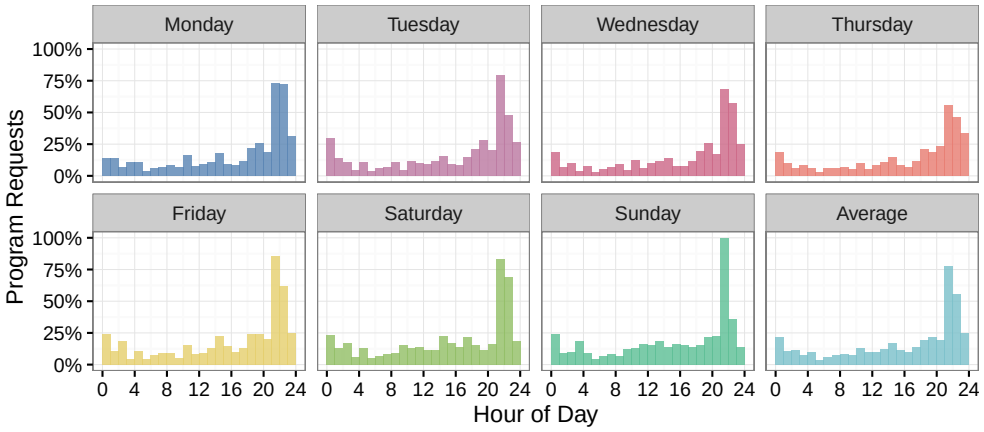


Figure 3.9: Original Airing: Day of Week and Hour of Day.

Program Requests Decay

Catch-up TV enables an *anytime* approach to content consumption that removes the time constraints associated with watching linear TV. Considering this new degree of freedom, the question of whether users take advantage and watch Catch-up TV content without regard to how long ago it was originally transmitted arises; thus, the purpose of this analysis is to evaluate the evolution of content relevance as it ages and new content is added to the Catch-up TV catalog.

Each program is classified according to its channel genre. The first observation of Figure 3.10 is that peak program demand occurs within 1 day of the original broadcasting time. A decrease in demand is observed due to night periods throughout. Furthermore, some genres completely dominate the number of content requests, namely *General*, *Movies and Series*, and *Kids*. *Entertainment*, *Documentaries*, *Sports*, and *News* genres quickly become irrelevant after the first two days.

Figure 3.11 enables a per-genre evaluation on the evolution of program requests with time, and shows that they exhibit very different decay patterns. Two main types of genres are clearly visible: those whose relevance quickly fades with time, such as *Sports*, *General*, and to some extent *Entertainment*; and others, whose relevance does not decrease so significantly with time, as is the case of *Documentaries*, *Kids*, *News*, and *Movies and Series* genres.

Given the time-sensitiveness of *News* programs, the results may seem odd; however, these TV channels are also known for hosting multiple sports and political debates, which might extend their overall time relevance.

As for the remaining genres, *Sports* and *General* have a high temporal relevance locality, while *Movies and Series*, *Documentaries*, and *Kids* programs do not typically exhibit any particular temporal importance, with the notable exception of some high impact TV series, such as Game of Thrones.

With more than 50% of the total program requests in the first day, and 79% after just 3 days, these results show that: on the one hand, the playback delay is a key factor on content popularity prediction, thus with the potential to be utilized in caching optimization algorithms, as demonstrated on Section 4.3; while on the other hand, increasing the Catch-up TV time window, from the current 7 days may not yield consumers any real benefit other than the psychological one of knowing that they have more content available, even if they will never watch it.

Even though the preceding analysis provides a deep insight into *how*, *when*, and *where* the service is utilized, with multiple perspectives on both content and users' characteristics, from a content delivery perspective, it is also important to understand how network traffic changes with time, and the potential gains achievable from smart caching and prefetching algorithms.

These different viewpoints are key to a properly planned and optimized CDN, in its various dimensions; therefore, being essential to the work presented on the next chapter.

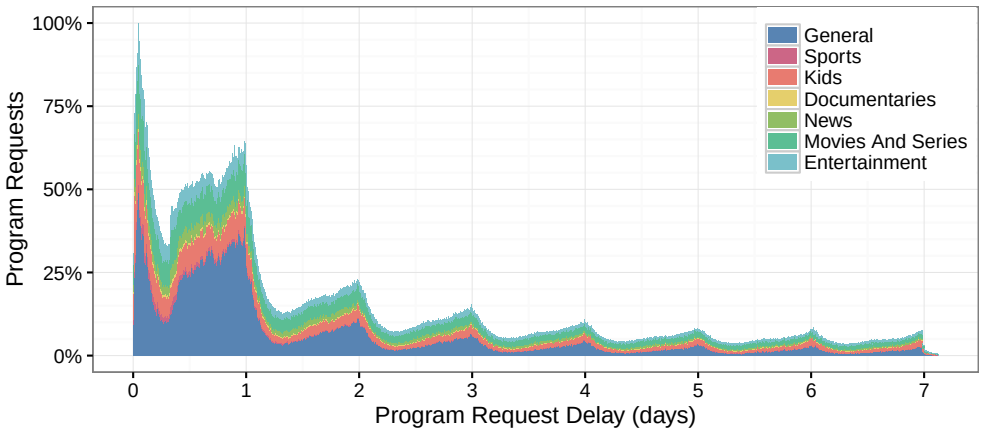


Figure 3.10: Total Requests vs. Request Delay.

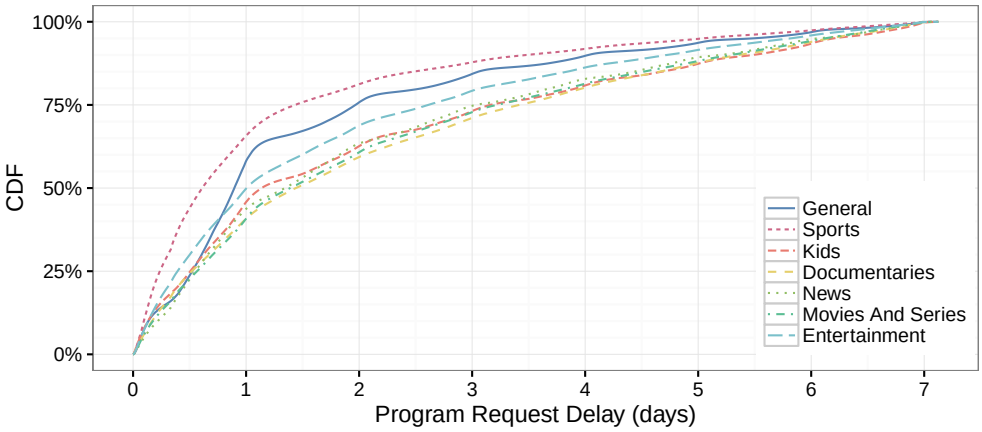


Figure 3.11: Total Requests vs. Request Delay CDF.

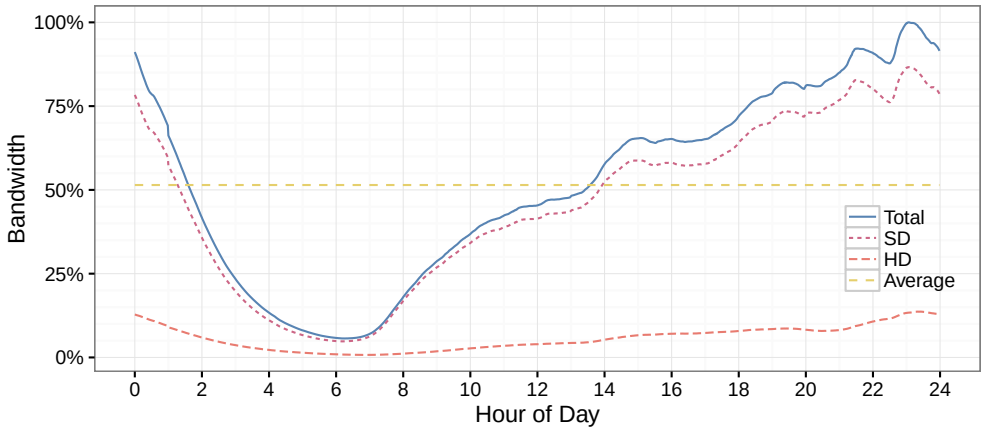


Figure 3.12: Bandwidth Consumption vs. Hour of Day.

Bandwidth Consumption

How much bandwidth is consumed, and how it varies between peak and off-peak hours is determinant in network capacity and investment planning, and to gauge the potential gains of network load distribution in time.

An accurate estimation on bandwidth consumption, which is a continuous measurement, must take into consideration not only the duration of each viewing session, but also the video quality of the requested programs.

Given that bandwidth measurements must take into account the active users at any given point in time, instead of the time of the program request, the viewing sessions data is expanded to provide information regarding the active programs at any given point in time, with a granularity of 1 minute. This granularity is chosen as a compromise between accuracy and the computational effort required to generate the data. Given that only a few programs have a duration of less than 1 minute, and that the programs' EPG-based durations have a resolution of 1 minute, we deem this approximation satisfactory.

In addition to knowing which programs are active at each point in time, information is also collected regarding the video quality. HD content is streamed at 6Mbps and requires exactly twice the streaming bandwidth of a SD content (3Mbps).

Figure 3.12 examines the variation of bandwidth demand with the hour of day, by averaging the bandwidth consumption data of the different days.

The explanation for the large gap in users watching HD and SD programs is threefold. The first reason is the lack of HD channels when compared to SD ones, since only 15.6% of the programs available on analyzed Catch-up TV service are HD. Second, the Catch-up TV user interface prioritizes SD over HD, which is an engineering design choice in order to reduce the overall bandwidth consumption as HD programs require twice as much bandwidth as their SD counterparts. Finally, because the vast majority of users are on DSL connections, with restrictions on bandwidth and amount of simultaneous video streams (the connection supports fewer HD streams than SD streams), users have an additional incentive to watch the SD versions in detriment of HD content.

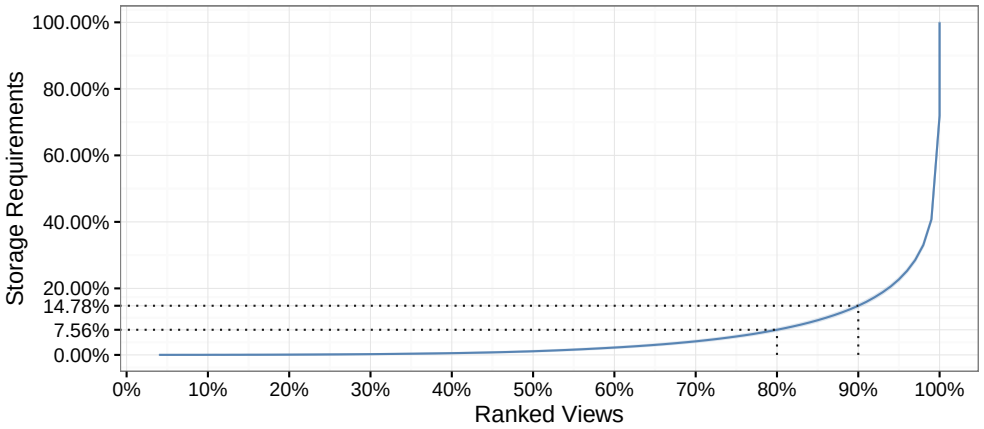


Figure 3.13: Required Storage Size vs. Top Program Requests.

Caching

This section is focused on metrics that provide insights on how cacheable Catch-up TV content is. To that end, an initial analysis of individual content popularity is conducted, in order to understand the theoretical limits of caching algorithms. Programs that are the most recurrently watched over their availability window present the best opportunity for caching improvements.

A study is conducted to determine how the cache storage requirements vary if they were to hold a given percentage of the most popular content. In this study, the programs available for request on each day are ranked according to their total number of requests.

While this approach does not take into consideration the impact of content locality, it does provide an overall perception on how cacheable Catch-up TV is.

The storage requirements are determined as a function of their duration and video quality. HD content, streamed at 6Mbps, requires twice the storage amount per unit of time than SD, which is streamed at 3Mbps. On average, 15.8TB of storage space is required to hold the complete set of available Catch-up TV programs regarding the service's 7 days window.

Figure 3.13 presents the average results for the 30 days analysis with the 95% Confidence Interval (CI) as a shaded region.

The first key observation is that the top 80% of programs only require 7.56% of the total storage requirements (∼1.2TB). The law of diminishing returns is clearly applicable, given than, for example, holding 90% of the most popular programs in cache would require a two fold increase in total storage requirements (∼2.3TB).

Caches requiring 1.2TB of fast storage, such as RAM, are well within the reach of common servers. It seems evident, then, that a properly designed caching mechanism, integrated in a CDN, and aware of the particularities of Catch-up TV content demand, would be able to show a stellar caching performance.

3.3.5 Conclusion

Considering the objectives set forth in the introduction, which stresses the importance of thoroughly understanding the Catch-up TV service usage in order to optimize the content delivery with benefits to operators and the consumers, it is clear that several highly valuable conclusions may be drawn from the research works. These major insights should be leveraged to improve the current delivery infrastructures and provide OTT Catch-up TV services that cater to users' needs efficiently, i.e. with reduced infrastructural requirements, OPEX and CAPEX, while providing an excellent QoE.

Users mostly prefer *General*, *Kids*, and *Movies and Series* content when watching Catch-up TV, whereas the remaining genres have a lower overall preference, in spite of the high availability of content in the less popular genres. The exploration of the most popular programs' characteristics shows that they were originally prime-time programs whose popularity was reinforced in Catch-up TV, hence proving the *superstar* effect of Catch-up TV, as opposed to the *long-tail* one. Furthermore, the results also show that users are very active throughout the day, particularly on weekends. The fact that Catch-up TV programs get over 75% of their total views in the first 3 days after airing, implies that expanding the Catch-up TV window from 7 days up to 14 or 30 days would not provide a real benefit to users, in spite of the added costs to Pay-TV operators.

From a content delivery perspective, the service optimization analyses revealed large differences between peak and off-peak bandwidth demand, which is problematic due to the average underutilization of network resources, which need to be dimensioned to approximately two times the average streaming bandwidth in order to avoid network bottlenecks. One possibility to ameliorate this issue would be to preload content on the client devices on low-demand hours, i.e. late night hours, in order to "flatten" the bandwidth curve, reducing the chance of network related issues on peak-hours, and improving the overall service quality.

Continuing on the topic of service delivery optimization, the caching-oriented study clearly shows that a small fraction of the programs are responsible for the vast majority of program request, and that caching the top 80% programs would only require 7% of the total corpus storage space.

In summary, all of these conclusions point to significant service improvement prospects, that can and should be used on next generation OTT multimedia CDNs to provide a better QoE to users, while simultaneously reducing Pay-TV operators costs. The exploitation of these opportunities is the target of Chapter 4.

Chapter 4

Improved OTT Delivery of Catch-up TV Services

Having framed the relevance of Catch-up TV services in the context of Pay-TV offerings, and the services' impact on users' habits, this Chapter focuses on leveraging that knowledge to improve the performance of Catch-up TV services delivered on OTT networks yielding four different, but related, scientific contributions.

First, given the importance of QoE on Over-The-Top multimedia services, the *QoE Assessment of HTTP Adaptive Video Streaming* [31] work specifically targets the challenge of modeling and estimating QoE in HTTP adaptive streaming scenarios.

The paper *Catch-up TV Forecasting : Enabling Next-Generation Over-The-Top Multimedia TV Services* [29] uses the available Catch-up TV dataset to evaluate the potential gains and advantages of demand forecasting for efficient delivery of Catch-up TV in OTT scenarios, while exploring several classes of machine learning models regarding their accuracy, computational requirement trade-offs, and deployability.

Next, *Over-The-Top Catch-up TV Content-Aware Caching* [28] proposes a content-aware cache replacement algorithm, Most Popularly Used (MPU), capable of taking advantage of content demand forecasts built using the previously evaluated machine learning models, to significantly outperform traditional cache replacement policies, such as LRU, LFU, and FIFO, and approach the optimal theoretical hit-ratio limits.

Finally, *Content-Aware Over-The-Top Delivery of Catch-up TV Services* [30] proposes, discusses, and provides an experimental evaluation of a content-aware delivery approach capable of leveraging online machine-learning techniques to continuously improve the performance of OTT delivery systems taking into consideration the content's characteristics and demand patterns.

4.1 QoE Assessment of HTTP Adaptive Video Streaming

Understanding how to properly model and measure QoE on OTT multimedia networks is an essential step before any actual optimization work, as a carefully calibrated QoE model facilitates an adequate evaluation of the benefits that users' may expect from the technical service improvements that will be presented in the upcoming sections of this chapter. Therefore, this section explores a novel approach to MOS estimations under HTTP Adaptive Streaming scenarios.

HTTP Adaptive Streaming (HAS) is a modern and increasingly popular approach to multimedia streaming on OTT networks. While traditional streaming protocols have been widely researched in the context of QoE, an all-encompassing model for QoE on HAS technologies that is able to consider aspects such as buffering events, initial playout delay and bit-rate change frequency, to name a few, is a necessity that must be addressed.

To that end, an objective analytical QoE model is devised with subjective calibration, respecting ITU-T recommendations, that is able to accurately estimate QoE on a variety of HAS scenarios. The full paper is included in *Appendix D*.

4.1.1 Introduction & Motivation

Regardless of the underlying technologies in OTT multimedia streaming delivery, a factor that has gained importance over the last years is that of Quality-of-Experience (QoE). QoE is a purely subjective metric, but it is so important that it can make or break the success of streaming service. It is heavily dependent on the underlying QoS parameters, but expands on QoS by taking advantage of human perceptions and focusing on the overall user experience.

HAS technologies aim to increase the users' QoE by embracing the natural variations of the underlying networks' performance, along with different terminal characteristics, while taking advantage of the ubiquitous HTTP infrastructure. The technology has gained traction with several implementations, including Microsoft's Smooth Streaming and MPEG-DASH, which are described in detail on Section 2.4.5. Given the characteristics of these adaptive streaming technologies, previous QoE estimation models do not directly apply, as they fail to encompass the new dynamics of a users' viewing session.

4.1.2 Scientific Contributions

This section presents the proposed model for accurate QoE prediction on adaptive HTTP streaming so that proper OTT streaming service performance monitoring might be conducted. The key scientific contributions are summarized as follows:

- Proposal of QoE estimation model considering the dynamic nature of HAS and human memory;
- Calibration of the proposed QoE model using industry-standard frameworks;
- Simulation and experimental results abiding to ITU-T P.1202 recommendations for non-intrusive bit-stream assessment of video media streaming quality.

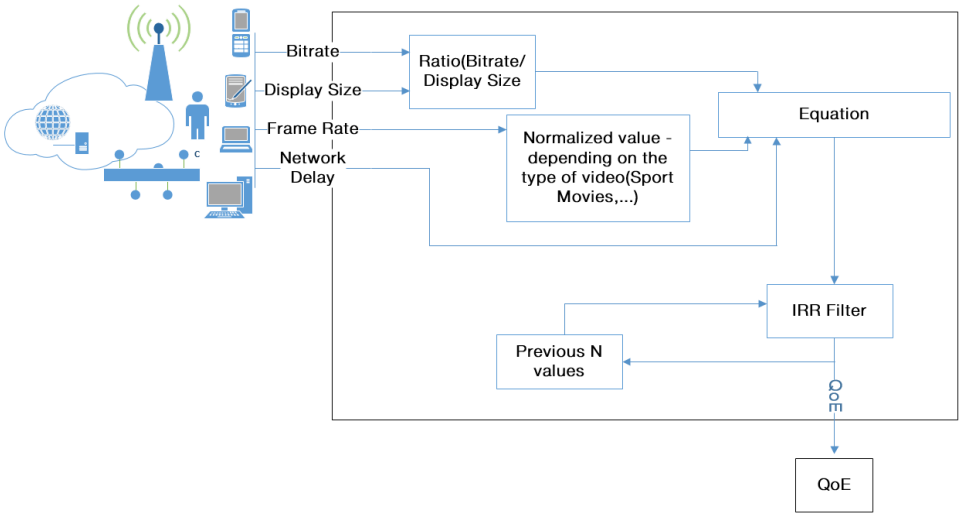


Figure 4.1: Adaptive HTTP Video Streaming QoE Estimation Architecture.

4.1.3 Adaptive QoE Estimation Model

As the overall experience of a video streaming session up to a given instant is influenced by the previous instants, the model needs to consider a memory effect over the elapsed period. The proposed algorithm may be decomposed into two phases, illustrated in the building blocks of Figure 4.1.

A first one, represented by Equation 4.1, classifies video chunks individually by considering the video's codec information, the client's terminal characteristics, and the network's QoS parameters in order to establish a baseline instantaneous MOS estimation, which is calibrated against the industry-standard OPTICOM's Perceptual Evaluation of Video Quality (PEVq) software [299]. *Bitrate* represents the H.264 encoded video bit rate in bits per second; *Fps* is the number of frames-per-second on the current chunk's video; *Rebuffering* corresponds to the buffering time in milliseconds; and *ScreenRatio* is the ratio between the device's screen area and the video's screen area - the areas are represented in squared-pixels.

$$
\begin{aligned}
v_1 &= 2.038 & v_2 &= 1.027 & v_3 &= 1.42^{-6} \\
v_4 &= 0.3031 & v_5 &= 3.064 & v_6 &= 0.5407 \\
v_7 &= 0.05652 & v_8 &= 1.756
\end{aligned}
$$

$$
\begin{aligned}
Score_{chunks} = & v_2 \arctan\left(Bitrate \times v_3\right) \times \log\left(v_4 \times Fps\right) \\
& - \log\left(v_5 \times Rebuffering + v_6\right) \\
& - \log\left(v_7 \times ScreenRatio + v_8\right) + v_1, \qquad (4.1)
\end{aligned}
$$

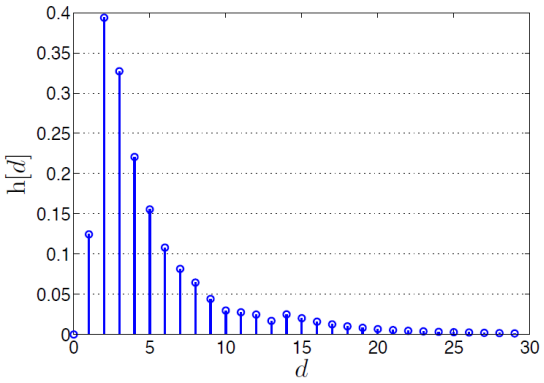


Figure 4.2: The impulse response of the memory filter in the first 30 seconds [11].

The second phase builds on the basic quality scoring of video chunks performed on phase one, with respect to their individual MOS estimates, and considers the impact of the previously reproduced chunks in the current MOS, emulating the human memory, and providing a QoE estimation that is a better approximation of the users' opinion of the service quality. The IIR response filter emulating human memory is presented on Figure 4.2 for a maximum duration of 30 seconds.

4.1.4 Main Results

To validate the proposed estimation methodology, a survey is conducted using real users to assess their quality of experience when watching variations of 2 reference videos: an animation one and a sports one. The videos are available in a set of 20 streams, using Microsoft Smooth Streaming with different sets of bit-rates per stream.

ITU recommends that questionnaires should have at least 50 responses in order to have enough confidence in the results; hence, we considered 64 users assessing the quality video streams available on a web page. Each video stream is classifiable with a MOS score, ranging from 1 to 5. In practice, however, it is difficult to get an average MOS higher than 4.5 or lower than 1.5, because not everyone classifies their experience with the extreme values of 5 or 1.

Figure 4.3 shows the results of the questionnaires, indeed demonstrating that the users' MOS estimate does not present values near the extremes – equal to 5 or 1.

The results show that MOS estimates produced by the survey are in line with the estimates provided by the QoE model, especially in the case of animation video streams (scenarios 15 to 20). In scenarios 1 to 15 the reference is the sports video, and the QoE model does not perform as good as it does in the animation video. This is likely an effect of sports videos, whose picture quality is harder to estimate due to fast moving scenes.

Overall, it is possible to conclude that the proposed model is able to closely track the subjective results, and does not present results near the extremes.

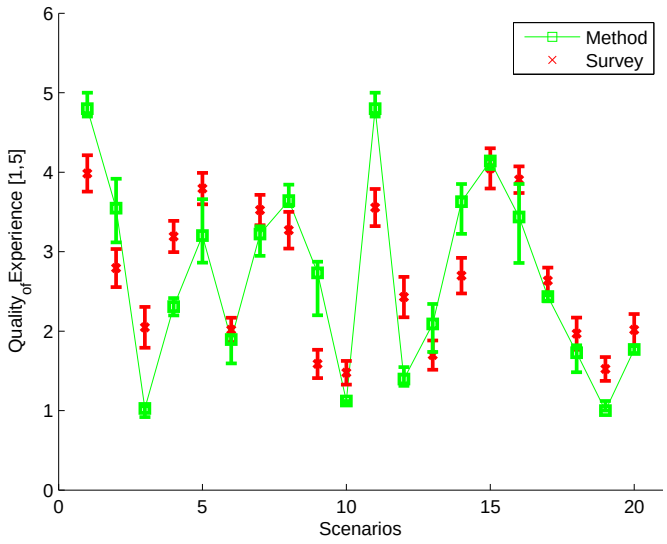


Figure 4.3: Survey with 20 Scenarios.

4.1.5 Conclusion

An advanced QoE model is proposed that is able to provide an accurate MOS estimation under adaptive streaming scenarios. The all-encompassing approach taken while developing the proposed model enhances the current state-of-the-art by demonstrating the incorporation of key characteristics of adaptive HTTP streaming in the estimation of the users' QoE, such as content bitrate, buffer and rebuffering delays, screen sizes, and Frames per second (Fps), in addition to contemplating the impact of previous users' experience in the form of a MOS human-like memory filter.

The results of a subjective assessment, using questionnaires according to ITU-T recommendations, are presented and shown to produce results that are a close match to the model's estimates.

This QoE model provides a cornerstone for a QoE-oriented analysis of HAS services delivered through OTT CDNs, allowing for performance metrics that take into account not only purely technical aspects, but also the subjective overall user experience.

The upcoming sections focusing on technical improvements of the OTT Catch-up TV delivery take advantage of the presented QoE evaluation framework to present a quantitative MOS assessment of the benefits that they provide.

4.2 Catch-up TV Forecasting : Enabling Next-Generation Over-The-Top Multimedia TV Services

To evaluate the potential gains and advantages of demand forecasting for efficient delivery of Catch-up TV in OTT scenarios, this research work explores several classes of machine learning models regarding their accuracy, computational requirement trade-offs, and deployability. The training process relies on a dataset comprised of Catch-up TV usage logs obtained from a Pay-TV operator's live production IPTV service containing over 1 million subscribers, which is studied in depth on Section 3.3.

A predictive and dynamic resource provisioning approach is proposed and evaluated in terms of bandwidth and storage savings. *Appendix E* contains the full publication.

4.2.1 Introduction & Motivation

A keystone in OTT multimedia services, which, if not properly accounted for, severely limits the systems' scalability and end-users' QoE, is CDN infrastructure optimization in its many aspects, ranging from caching optimization, bandwidth reservations, Point-of-Presence (PoP) location, and elastic resource provisioning to cope with varying demand [146]. While static optimization is possible, by thoroughly analyzing past demand data, it is error prone and subject to human-error. A more interesting scenario with potentially higher efficiency gains is that of autonomic and dynamic CDN optimization, capable of providing better resource usage, lower costs, and power consumption; however, this dynamic approach is rife with difficulties and is accompanied by a crucial obstacle: the need to accurately forecast demand in a practical time frame.

This necessity is addressed in this research work, which creates and evaluates forecasting models suitable for being employed as part of a solution for cloud resource orchestration in CDNs [146, 300], following a step-by-step approach to ensure clear, reproducible, and sound results that may be used in subsequent research efforts and applied to existing or new CDNs. In order to create, assess, and propose feasible and accurate forecasting models for Catch-up TV content consumption, a predictive and dynamic resource provisioning approach is proposed and evaluated in terms of bandwidth and storage savings. The attained results show that the forecasting models are able to produce accurate bandwidth and storage requirements forecasts, which may be used to achieve considerable power and cost savings.

4.2.2 Scientific Contributions

The key scientific contributions achieved in this work are outlined below:

- Definition of a step-by-step approach for Catch-up TV demand forecasting;
- Proposing the TASE metric, a new approach for comparing the performance of different predictive algorithms;
- Benchmarking several classes of machine learning algorithms in the context of Catch-up TV forecasts;
- Bandwidth and storage requirements evaluation using the predictive models.

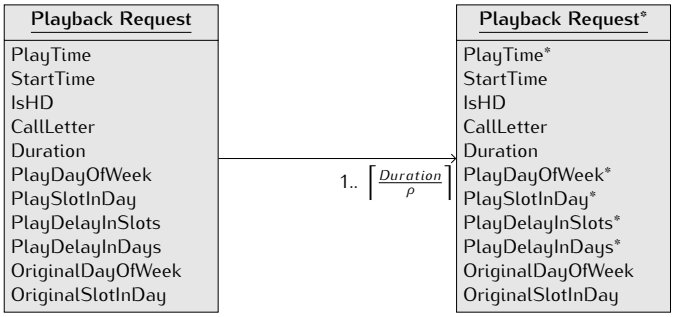


Figure 4.4: Playback Requests Mapping into Continuous Sessions.

4.2.3 Preliminary Data Analysis & Strategy

In order to understand how to build forecasting models, it is first necessary to gain a deep insight on the data that is used to create the required machine learning algorithms. To that end, the seminal work presented in Section 3.3 is essential in clearly hinting at the existence of demand patterns with variations throughout the hour of day and day of week. However, in spite of suggesting that patterns exist, no information is provided on what data features are most relevant in determining their behavior; therefore, a preliminary data analysis with a focus on demand forecasting is key.

Given that the dataset only contains records of playback start events, not reflecting the continuous nature of video playback, one of the first conclusions is the need to create an expansion of these events to reflect their time-continuity for the total duration of the video content. This effect is especially important in Catch-up TV scenarios where content is most of the time continuously streamed for its entire duration [52, 28].

The discrete events must, therefore, be able to reflect this continuity and are replicated to simulate a set of periodic request events up to the total duration of the content. In practice, additional requests are introduced to spread the content over the considered time-slots. Figure 4.4 illustrates this mapping, which has a significant impact on the data dimensionality. Fields marked with an asterisk ($*$) are recomputed to take into account the continuity expansion.

Given the insight on the available features and the demand forecasting target, the issue remains on establishing an adequate strategy for forecasting, especially considering that *IsHD* and *CallLetter* are categorical predictors - that can only take on one of a limited set of values - and the fact that the dataset consists of millions of samples.

A design decision is made to split the forecasting problem per individual channel. This proposition has two main advantages: first, there is no need to convert the *CallLetter* predictor into dummy variables, which would increase the data dimensionality; second, because the prediction is performed for each individual channel, we expect a higher accuracy per channel. This approach does, however, increase the risk of model over-fitting which must be considered.

A high level step-by-step strategy is illustrated in Figure 4.5. This process follows industry guidelines for data mining processes, i.e the CRISP-DM [248].

Figure 4.5: High Level Forecasting Strategy.

Forecasting Models

Considering the goal of forecasting content demand, a subset of commonly used regressive machine learning models are chosen as representatives of the main predictive regression models' classes:

- *Bayesian Regularized Neural Networks (BRNNs)*: a class of NNets [301, 302];
- *Random Forests (RFs)*: classification and regression based on a forest of trees using random inputs [303];
- *k-Nearest Neighbors (k-NNs)*: widely used in classification and regression [304];
- *Partial Least Squares (PLS)* regression [305];
- *Support Vector Machine (SVM)* with a Radial Basis Function Kernel [306];

4.2.4 Pre-Processing & Feature Selection

The dataset predictors exhibit different scales, standard deviations, and average values. These discrepancies in scale and statistical properties often impair the numerical stability and bias of learning algorithms, potentially favoring some predictors over others, not because of their real importance but because of their different scales.

In order to compensate for these discrepancies and treat every predictor as equal inputs to learning algorithms, it is important to *scale*, *center*, and correct the *skewness* of each predictor as described on Section 2.7.2. In this work, individual predictors are corrected for skewness using a Yeo-Johnson transformation [307], centered, and scaled.

In addition to data pre-processing, feature selection is also crucial in properly tuned machine learning models, especially as data dimensionality grows. Having less features to measure or acquire may not only improve the performance of predictive algorithms but also reduces computational and data acquisition costs. Also, models using less predictors are typically more interpretable and better able to adjust to unknown predictors.

The two broad classes of feature selection encompass *supervised* and *unsupervised* methods, depending on whether a result or outcome variable is used in the feature selection process – *supervised* or *unsupervised*, respectively. Each approach is discussed in detail on Section 2.7.3.

An unsupervised technique that is useful for identifying relevant predictors is the predictors' cross-correlation. If a dataset contains predictors that are highly correlated ($\rho > 0.95$) there is a good chance that these predictors convey the same information; hence, one of them is a good candidate for disposal. The cross correlation plot of the available predictors is shown in Figure 4.6.

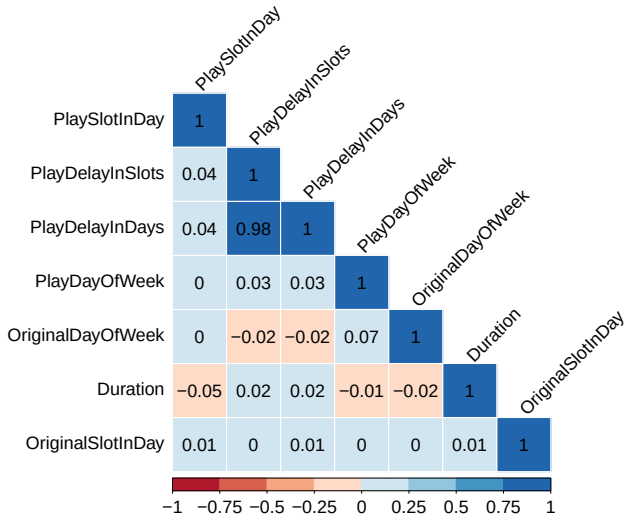


Figure 4.6: Unsupervised Feature Selection - Cross-Correlation.

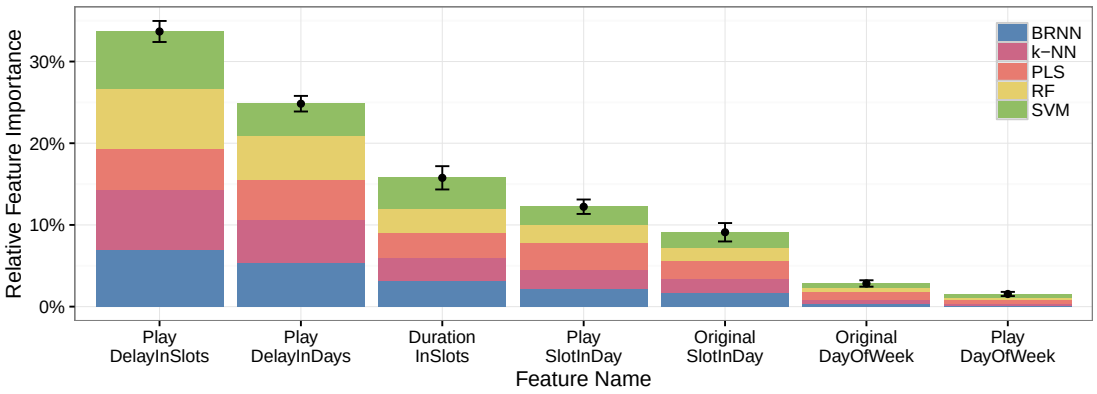


Figure 4.7: Ensemble Feature Selection - Weighted Relative Feature Importance

---

For supervised feature selection, an *ensemble* [259] approach is taken relying on the *fscaret R* package [308]. The results of Figure 4.7 are generated by running the supervised feature selection algorithm using each individual TV channel data, averaging out the final results, and computing the 95% CI.

Considering the results of the unsupervised and supervised feature selection procedures, it is possible to make a decision on the predictors that should be removed before training the regression models.

From the unsupervised feature selection results, the *PlayDelayInDays* is chosen for removal, while from the supervised selection perspective, the *OriginalDayOfWeek* and *PlayDayOfWeek* features should be ignored. A decision is made to take both results into consideration and remove these 3 predictors; thus, the final set of predictors comprises: *PlayDelayInSlots*, *Duration*, *PlaySlotInDay*, and *OriginalSlotInDay*.

4.2.5 TASE Prediction Performance Measurement

Even though a common indicator of prediction accuracy is RMSE, which is an error measure of the distance between predicted and observed values, the fact that it is scale-dependent [309, 310] makes it unfit for comparing the performance of forecasting models for different TV channels, which exhibit very diverse demand volumes.

Some of these limitations are addressed by Mean Absolute Scaled Error (MASE) [309, 311], in the context of time-series forecasts, as this indicator's scale-free properties enable an accurate comparison of different forecasting algorithms. MASE is composed of two main parts (Equation 4.2): the numerator computes the average absolute prediction error e_t; the denominator, the *scaling factor*, scales this error with the Mean Absolute Error (MAE) assuming naïve forecasting. Both the numerator and denominator share the original data's scale; hence, MASE is scale-free. In this equation, n is the total number of samples to forecast, while Y_i represents the naïve forecast for period i.

$$\text{MASE} = \frac{\frac{1}{n}\sum_{t=1}^{n}|e_t|}{\frac{1}{n-1}\sum_{i=2}^{n}|Y_i - Y_{i-1}|} \tag{4.2}$$

Given that MASE was developed to target time-series forecasts, where a natural order exists between the different samples, the scaling factor is easily computed through the naïve forecasting approach. However, in our scenario, no such order exists, and the scaling factor, as defined in MASE, is not appropriate for scaling the average absolute prediction error. To compensate for this shortcoming, the scaling factor in MASE is replaced by the average outcome of the training set, as per Equation 4.3, and denominated Training Average Scaled Error (TASE). In the proposed TASE metric, the numerator averages the absolute prediction error e_t for the n total forecasts, while the denominator scales it with the average value of the m training samples. Per the same principle of MASE, TASE exhibits scale-free properties and is used in the evaluations as a key performance indicator.

$$\text{TASE} = \frac{\frac{1}{n}\sum_{t=1}^{n}|e_t|}{\frac{1}{m}\sum_{i=1}^{m}|Y_i|} = \frac{m\sum_{t=1}^{n}|e_t|}{n\sum_{i=1}^{m}|Y_i|} \tag{4.3}$$

4.2.6 Model Building Methodology

Before delving into the actual model building and performance testing phase, it is first necessary to establish the tests' conditions and assumptions.

The tests are implemented in *R* [312] using RStudio [313], and run on a VM with 2 Intel E5-2640v3 CPUs (32 cores), and 64GB of RAM.

Even though the performance of the models considered in this analysis are dependent on their actual implementations, the tests are all performed in identical conditions and use reference and commonly used libraries and implementations of the predictive models.

The complete Catch-up TV dataset contains 30 days of requests logs. A decision is made to split the dataset into two groups. The first comprises the initial 23 days and is

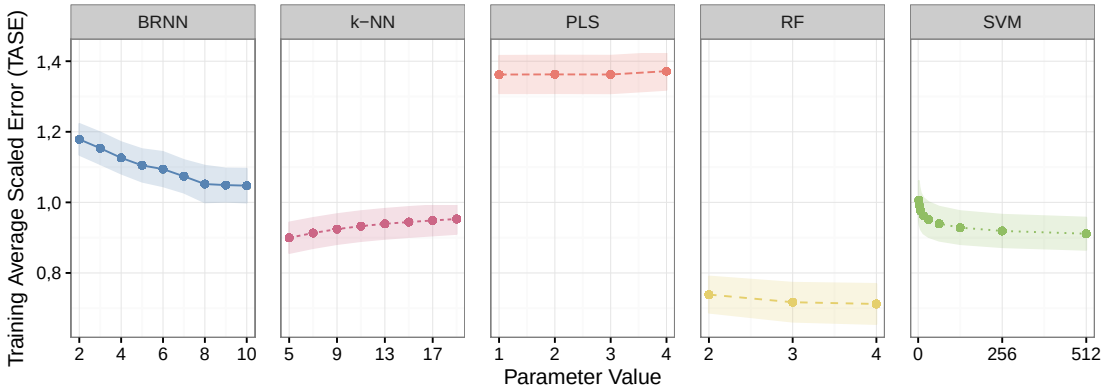


Figure 4.8: Tuning Parameter Selection.

used to train the model, while the second relies on the last 7 days and serves to prove that the training process has a good generalization ability in the face of unknown data.

The training phase relies on 10 times repeated 10-fold cross validation resampling. This particular cross-validation process is selected because it has been shown to produce similar results as the more computationally burdensome Leave-One-Out Cross-Validation (LOOCV) approach [262]. This and other resampling approaches are described in depth in Section 2.7.4.

4.2.7 Main Results

Hyperparameter Tuning

Before proceeding with performance testing, it is first necessary to find adequate tuning parameters for each model. Properly tuned models are essential to produce good results. Each model type has its own set of tune parameters that must be adequately configured to maximize their performance.

In order to determine suitable parameters for each predictive algorithm, a grid search is conducted. The grid search is an hyperparameter optimization approach whereby an exhaustive search is conducted using a set of manually specified parameters to determine which parameter combination yields the best model performance, according to the previously defined TASE metric. These tuning parameters — or variables — are different per model, and must obey to distinct constraints. To reduce the chance of over-fitting, cross-validation is performed using 10 times repeated 10-fold cross validation. A maximum of 10.000 samples are selected from each channel's training data.

Figure 4.8 presents the grid search results by individual model, from which several conclusions may be drawn.

Starting with BRNNs, where the tuning parameter is the number of *neurons*, it is clear that its performance improves with the addition of up to 8 neurons, after which the models' performance stabilizes. As for k-NN, the results show that the cost of generalization, translated into a higher number of neighbors (k), is a decline in global

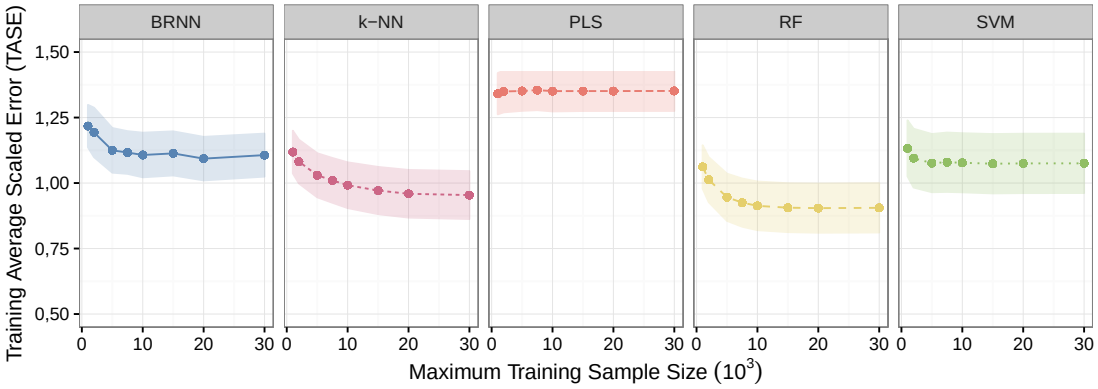


Figure 4.9: TASE Scaling with Training Sample Size.

performance; hence, k is set to 5 for the forthcoming performance evaluations. PLS models have a completely different behavior and are shown to perform roughly the same regardless of the chosen number of components. The adequate number of components, *ncomp*, selected for the model training phase is 3. In this tuning phase, RFs show the best performance of the considered models, especially when using the maximum number of *mtry*, 4. Finally, SVMs' performance improves with an increase in cost, which stabilizes for metric costs over 128. The best value of 512 is chosen for C.

TASE Scaling with Training Sample Size

This scaling analysis focuses on TASE, described in Section 4.2.5, and is presented in Figure 4.9. As previously mentioned, TASE provides a scale-free error metric that is suitable for comparing the performance of the forecasting models between different TV channels, which exhibit distinct demand profiles and scales. The lower the TASE, the better the prediction, with 0 corresponding to a perfect forecast, i.e. the predictions match the observations.

The first observation is that the models' performance with respect to TASE does not appear to vary significantly with training set sizes greater than 10.000 samples.

When considering each model individually, additional conclusions may be withdraw. PLS provides the worst results, which is expected due to the model's simplicity, when compared with alternative approaches. BRNN fares significantly better than PLS, but worse than the other competing models. SVM's performance appears to be somewhat insensitive to maximum training sample sizes over 2.000, which is remarkable as it fares better than BRNN. The clear winner is RF, whose performance improves greatly from small sample sizes up to 10.000 maximum training samples, after which the performance gains are reduced. Lastly, k-NN provides a middle-ground performance between RF and SVM, especially for higher maximum training sample sizes.

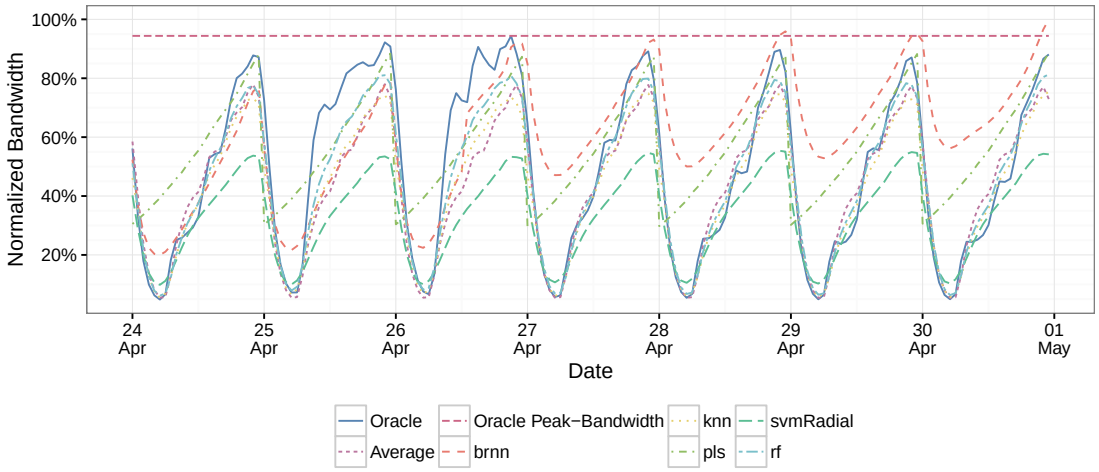


Figure 4.10: Bandwidth Requirements Forecast.

Bandwidth Forecasting

The issue of bandwidth requirements is prevalent in computer networks, especially in those dedicated to bandwidth-intensive multimedia streaming. To understand how they vary with time, Figure 4.10 provides a comparison of the *Oracle*, *Average*, and predicted bandwidth demand per forecasting model. An additional helper line, *Oracle Peak-Bandwidth*, is added to represent the maximum observed bandwidth during the forecasting period.

The resuls of Figure 4.10 showcase the models' demand forecasting abilities. Random Forests (RFs) and k-Nearest Neighbors (k-NNs) exhibit the best overall performance, closely tracking the observed bandwidth requirements and improving over the *Average* demand baseline. Bayesian Regularized Neural Networks (BRNNs) and Support Vector Machines (SVMs) provide a lesser approximation of the bandwidth demand curve, with BRNN sometimes over-predicting demand for the late night hours, and SVM consistently under-predicting peak-demand. As expected, Partial Least Squares (PLS) provide the worst approximation. Overall, except for PLS, all forecasting models are able to provide a demand forecast that approximates the observed demand.

To complement these demand forecast results, Figure 4.11 provides an analysis on the cumulative bandwidth requirements. This point-of-view allows a better insight on the potential power and cost savings.

Oracle Peak-Bandwidth provides an upper bandwidth limit and corresponds to static provisioning at maximum capacity; *Oracle* presents the actual bandwidth demand; *Average* represents the average bandwidth demand according to historical data; finally, the prediction curves per machine learning model are presented.

The results indicate that less than 50% of the total statically provisioned resources, *Oracle Peak-Bandwidth*, are required to address the dynamic demand; therefore, an ideal

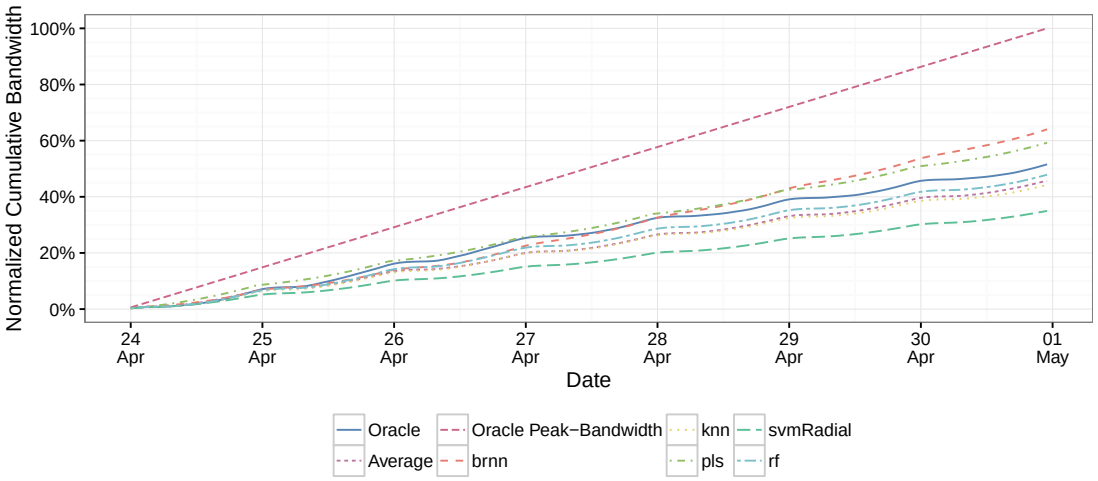


Figure 4.11: Bandwidth Savings.

resource provisioning system, with a linear relationship between power consumption and cost, would be able to use less than 50% of the total power, and reduce more than 50% of costs. In practice, the actual relationship between provisioned resources, cost, and power consumption is not this simple, but these results provide a ball-park indicator of the potential savings.

Storage Forecasting

Other application of demand forecasting systems for Catch-up TV is storage optimization. Previous studies have shown that users do not take advantage of the complete content catalog at their disposal (Appendix C), leading to wasted storage.

To address this issue, an analysis is conducted with the purpose of assessing, at each time of day, the programs actually requested by users and their storage requirements. These requirements are determined as a function of the content's duration and video quality. For the considered Catch-up TV service, HD content requires twice the storage amount per unit of time when compared to the SD counterpart.

Figure 4.12 conveys the results of this investigation and shows that, similarly to the bandwidth analysis, significant gains are achievable by considering demand forecasts.

The *Available* curve reflects the total storage required to hold the complete Catch-up TV content catalog; the *Oracle* curve shows the storage requirements of the actually requested content; the *Average Storage* curve is presented to reflect the static analysis over historical data; lastly, individual curves are shown per machine learning model.

The slight variations in the *Available* storage requirements curve are due to Catch-up TV content being added and removed from the content catalog throughout the day.

The forecast results vary according to the underlying machine learning model used, with RF providing the most accurate results, and PLS the worst. All machine learn-

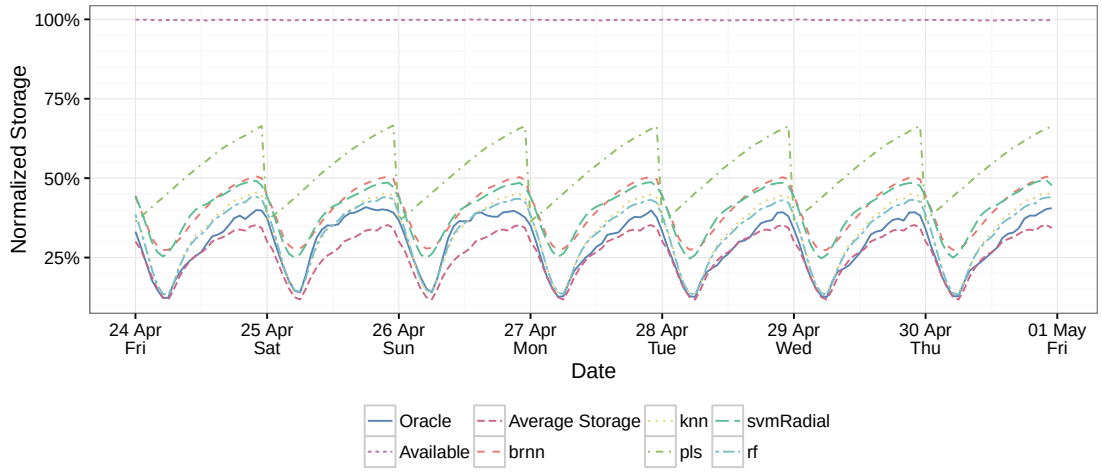


Figure 4.12: Storage Savings.

ing models slightly overestimate the actual storage requirements. In spite of the overestimation, it is possible to observe that the storage requirements correspond to a fraction of the content catalog, peaking at under 50% of the total *Available* catalog.

4.2.8 Conclusion

Considering the results, it is possible to observe that the predictive models are accurate enough to produce usable bandwidth and storage requirements forecasts to be used in dynamic operational environments. RF and k-NN produced the most accurate predictions and surpass the performance of static historical analysis.

Significant bandwidth and storage savings are possible in dynamic provisioning environments, leading to potentially large savings on cost and power consumption. The results provide an indication that the service performance should not be affected, as it ensures that there are enough available resources to meet the demand. However, the actual service performance, from a QoE perspective, is very dependent on statistical aspects of user demand, particularly in terms of its variance during the forecasting time-slot. To compensate for these factors a slight over-provisioning may be required to prevent spurious QoE drops and ensure a smooth user experience.

The chosen time-slot duration of 1 hour proves to be adequate for generating accurate forecasts. Shorter forecasting time-slots are possible, but are constrained by computational requirements, in spite of their potential for even more dynamic adjustments.

Additional applications of the forecasting models are presented on the ensuing sections, starting with a showcase of the demand forecast models' suitability for caching performance optimization on OTT CDNs, and proceeding with a content-aware OTT delivery architecture that heavily relies on the demand forecasting abilities to improve the end-to-end service performance, from a technical and QoE-centric perspective.

4.3 Over-The-Top Catch-up TV Content-Aware Caching.

The migration of popular Catch-up TV services to modern Over-The-Top (OTT) multimedia delivery infrastructures creates a wide set of scalability challenges which are commonly addressed using CDNs relying on caching nodes close to users. Given that the overall performance of CDNs is highly dependent on the efficiency of caching nodes, measured in hit-ratios and upstream bandwidth savings, and that the modification of caching algorithms is a feasible operation in commonly used proxy cache solutions, such as ATS [75], Nginx [77], or Varnish [78], this work focuses on improving this crucial component. The complete paper is included in *Appendix F*.

4.3.1 Introduction & Motivation

The use of general-purpose caching nodes is not optimal as it does not consider the particularities of Catch-up TV content, namely its dynamic popularity behavior, superstar effects, and relevance decay, as shown in existing scientific literature [139, 52] and the previous work presented on Section 3.3 and Appendix C. Since caches are limited in size and are relatively small when compared to the whole catalog of available Catch-up TV content, which may contain tens of thousands of TV programs, it is crucial to make the most out of the available resources.

Improving caching performance requires taking into consideration the underlying content demand patterns, and properly exploring them; therefore, this work proposes a novel caching algorithm, Most Popularly Used (MPU), that takes advantage of content demand forecasts produced by a predictive machine learning model, as specified in Section 4.2, to improve its caching decisions considering specific characteristics of the Catch-up TV content requested, i.e. in a content-aware manner. In addition to MPU, two additional caching algorithms that are forecast-aware are also implemented and tested – LRU-Weighted (LRU-W) and LFU-Weighted (LFU-W) – to further demonstrate the applicability of demand forecasts to improve existing caching algorithms.

To ensure the soundness of the proposed approach and its results, the Catch-up TV request logs previously characterized in detail on Section 3.3 and Appendix C are used to replicate users' requests. The results show that content-aware approaches are suitable for significantly improving existing CDN caching nodes, and that their computational implementation cost is comparable to that of commonly used algorithms.

4.3.2 Scientific Contributions

Considering the need for better caching algorithms in OTT Catch-up TV delivery scenarios, this work provides the following novel contributions:

- Proposal of a content-aware cache replacement algorithm, Most Popularly Used (MPU), capable of taking advantage of content demand forecasts;
- Thorough simulation-based evaluation of the proposed algorithm considering relevant, industry-standard, competing alternatives.

4.3.3 Most Popularly Used (MPU)

MPU leverages content demand knowledge to make cache replacement decisions based on "priority maps". Priority maps are generated by online predictive machine learning algorithms, whose responsibility is to produce accurate content demand forecast for a given period. The predictive models are continuously improved by using past data. The generated priority maps contain enough information to unequivocally identify Catch-up TV items and their expected number of requests at each point in the future.

MPU cache eviction policy favors items that have a greater expected priority, in detriment of others with lower expected priorities. In order to properly depict the inner workings of MPU, we assume that a cache system containing a list $\mathscr{C}$ exists capable of holding S elements, and that the items to cache are represented by the set $\mathscr{I} = \{i_1, i_2, i_3, i_4, i_5 ... i_n\}$ and have an associated numeric priority from the set $\mathscr{P} = \{p_1, p_2, p_3, p_4, p_5 ... p_n\}$, so that item i_1 has p_1 priority, and so forth. $\mathscr{H}$ is a counter registering the total number of hits, while $\mathscr{M}$ counts the total number of misses.

These steps summarize how MPU works when an item is requested:

1. If the item already exists in cache, it is returned to the caller, and the total hit count is incremented;
2. If an item does not exist in cache, a miss is registered and the item is fetched from the origin server so that it may be returned to the caller;
3. If the cache is full or if a newly fetched item has a priority higher than the lowest priority in cache, MPU removes the lowest priority item and inserts the new one.

The pseudo code of MPU is presented in algorithm 1.

Algorithm 1: Most Popularly Used Algorithm

Input: $\mathscr{I}, \mathscr{P}$
Output: $\mathscr{H}, \mathscr{M}$
For every item $i \in \mathscr{I}$, perform the following operations.
Case 1: if $i \in \mathscr{C}$ then :
    *Checks if item i exists in cache, if so, increment the total hits;
    $\mathscr{H} \leftarrow \Delta 1$;
Case 2: otherwise, if $i \notin \mathscr{C}$ then :
    *New miss is registered and the item is fetched from the origin server;
    $\mathscr{M} \leftarrow \Delta 1$;
Case 3: if $|\mathscr{C}| \geq S$:
    *Cache is full. Checks if new item i has higher priority than lowest
    *priority item in cache;
    if $p_i > \mathscr{C}_{min(p)}$:
        *Delete the item with lowest priority in cache ;
    *Insert new item i in the cache $\mathscr{C}$;

4.3.4 Testing Methodology

The tests are implemented in *R* [312] using RStudio [313], and run on a VM with 2 Intel E5-2640v3 CPUs, and 64GB of RAM. Even though the performance of the models considered in this analysis are dependent on their actual implementations, the tests are all performed in identical conditions and use common libraries.

The full dataset is presented in detail on Section 3.3. When pertinent, the 95% CI is shown on the average values' curve and data points, and the results are presented in a normalized fashion, ranging from 0% to 100%, to facilitate a graphical analysis.

Content Demand Forecasts and Testing Data

MPU relies on demand forecasts to make caching decisions; therefore, to conduct a performance evaluation it is necessary to build predictive models. The models are built according to the process described on Section 4.2.

The request logs are split into two separate groups: the first 23 days are used for training a Random Forest (RF) machine learning model; the remaining 7 days worth of logs, from April 24 up to April 30, represent the testing dataset and are used to create a sequential list of program requests which are inputs of the caching algorithms.

Reference Cache Algorithms

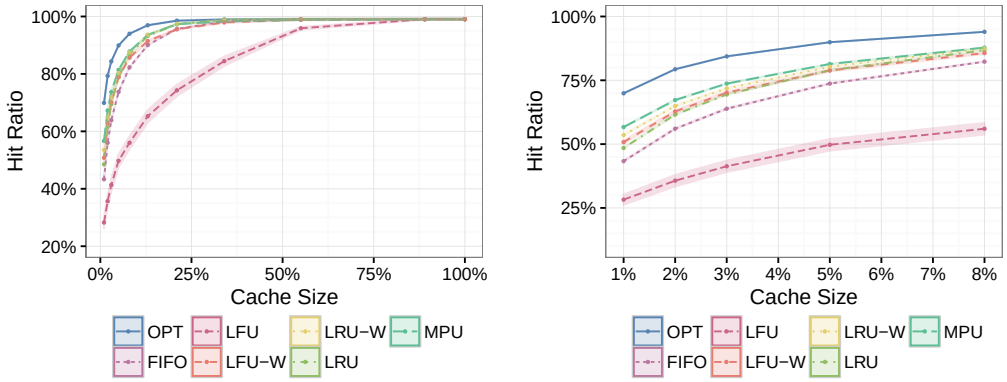
In addition to testing MPU, LRU-W and LFU-W, reference caching algorithms, LFU, LRU, and FIFO, are implemented and serve as a comparison base. Even though other caching algorithms exist, most are either variations or combinations of the aforementioned algorithms. Furthermore, to understand the upper limit of achievable hit-ratio performance, Bélády's optimal page replacement algorithm (OPT) [202] is also implemented. The algorithms' core implementations are kept as similar as possible.

Cache Sizing

In order to explore the effect of different cache sizes in the performance of each algorithm, and the associated cost-benefit trade-offs, the caches are sized as fractions of the total number of unique available programs. Therefore, a cache size of 100% corresponds to a cache with the ability to hold the entire content catalog available on the 7 days testing window. To simplify the caches' implementation, each program is assumed to require 1 storage unit.

4.3.5 Main Results

The results presented in this section are centered around two key metrics: *Hit-ratio* represents the ratio between the number of *hits* and the number of program requests, and is an indicator of how good the caching algorithm is on guessing programs that will be requested in the near-future; the *Run time* reflects the time required to run caching algorithms' code – the lower the better.



(a) Full Results.

(b) Zoom-In.

Figure 4.13: Hit Ratio vs. Cache Size.

Hit Ratio vs. Cache Size

This analysis explores the impact of different cache sizes in the overall caches' hit-ratios. The results are presented in Figure 4.13.

Starting with Figure 4.13(a) it is possible to observe that the optimal caching algorithm (OPT) always provides the best performance, which is to be expected, while MPU, closely followed by LRU-W and LFU-W, performs better than the remaining algorithms.

The best-performing traditional algorithm is LRU, which performs worse than MPU but much better than LFU and FIFO strategies. The algorithms' performance converges for cache sizes greater than 25%; however, we argue that this is not a common realistic scenario, which is mostly focused on cache sizes smaller than 10% of the overall corpus.

To better analyze this region, Figure 4.13(b) presents a zoomed-in plot of the same results, where a clearer comparative study may be conducted. In this figure, it is possible to observe that, for cache sizes of 1%, MPU provides a hit-ratio 17% higher than LRU, the best performing traditional caching algorithm. The results demonstrate that MPU may be used to either lower the caches' sizes, for a given target hit-ratio, or to improve the cache hit-ratios for fixed storage sizes.

Moreover, the remaining "forecast-aware" caching algorithms, LRU-W and LFU-W, consistently outperform the other traditional caching strategies, with LRU-W closely tracking the performance of MPU.

Hit Ratio vs. Time

Exploring how the caches' hit-ratios evolve with time is essential in Catch-up TV services where content popularity changes with time, and knowing the steady-state performance of caching nodes is a requirement. In order to perform this analysis, the cache sizes are set at 1% of the total program corpus, which was previously shown to be a data point providing a cost-benefit trade-off where good caching performance is achievable

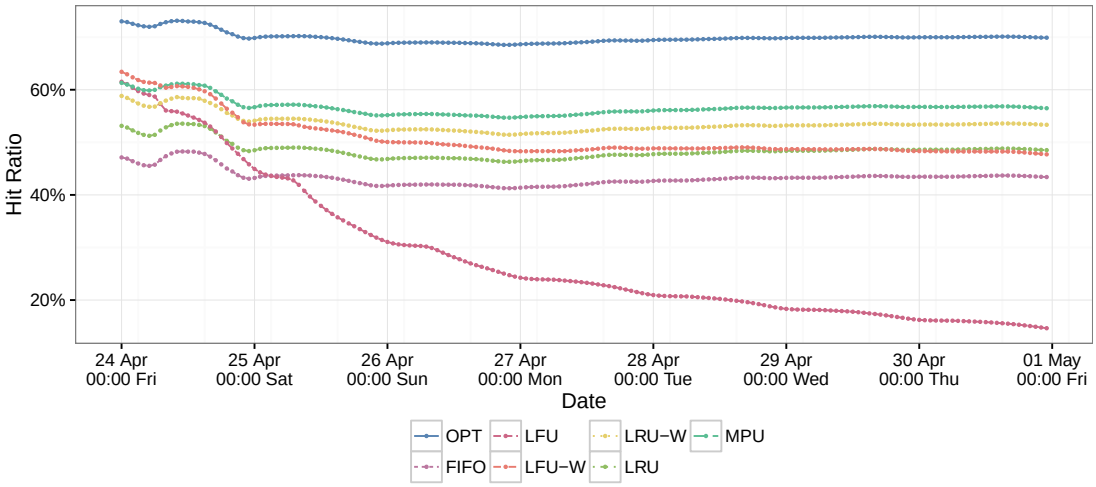


Figure 4.14: Hit Ratio vs. Time.

with less than an order of magnitude of the total content.

Figure 4.14 presents the time-varying hit-ratio results for each caching algorithm where it is possible to observe that, as time progresses, some algorithms adapt better than others to content requests. The ideal algorithm, OPT, provides the best overall caching performance, which is kept approximately constant with a 70% hit-ratio. MPU provides the second-best results, with a significant performance advantage over LRU-W and LFU-W, the next best performing algorithms. In spite of the different performance results, the overall hit-ratios of the caching algorithms follow a similarly-behaved curve that stabilizes after the first day and provides a consistent steady-state performance.

As for LFU, in spite of the excellent results for the early hours of day 24, its performance progressively diminishes with time, which might be explained by the effect of "cache pollution", whereby items that were initially highly popular, but lose relevance, prevent other newer items from populating the caches; hence, leading to low hit-ratios.

The small increase in hit-ratios on all algorithms in day 24 is believed to be due to accentuated users' demand for popular content in some times of the day, i.e. a result of the *superstar* effect which also happens in the remaining days, albeit at a smaller scale.

The evolution of caching performance with time shows that the "forecast-aware" caching algorithms, MPU, LRU-W and LFU-W, ensure the best hit-ratio performance.

Cache Run Time vs. Time

The final evaluation is centered around the evolution of cache run time with time. The cache sizes are set at 1%. This is an important metric, as the high-performance requirements of CDNs constrains the selection of caching algorithms to those that are computationally efficient and scalable. Figure 4.15 presents a graphical analysis on how the computational requirements of each caching algorithm varies with time. OPT is

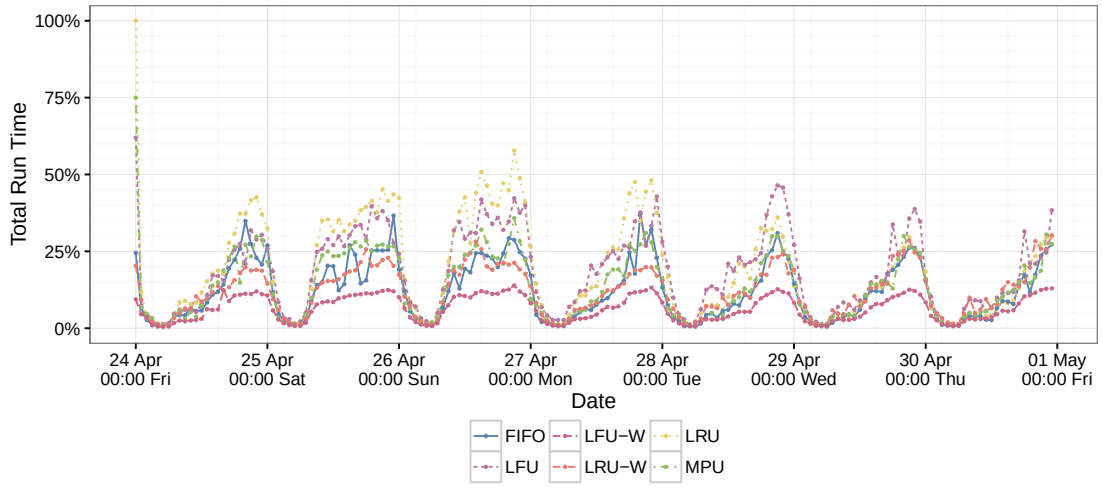


Figure 4.15: Cache Run Time vs. Time.

excluded from the results as it is not implementable in practice.

It is possible to observe that all caching algorithms exhibit a similar behavior with respect to their computational requirements, even though some algorithms do require more processing time than others. LFU-W is the least computationally demanding caching algorithm, followed by LRU-W, MPU and FIFO, while LFU and LRU require more time to perform their tasks. The initial observed run time peak for every caching strategy is due to the caches' warm-up process.

These results indicate that the addition of forecasting knowledge to caching algorithms not only aids with the hit-ratio performance but also allows for caching algorithms that are less computationally demanding.

4.3.6 Conclusion

Multimedia delivery in OTT environments is particularly challenging, specially in the case of Catch-up TV content with its dynamic demand patterns that make it hard for traditional caching algorithms to exhibit, and sustain, high performance levels.

To address the issues with Catch-up TV caching in OTT environments, a novel algorithm, MPU, is proposed that is able to leverage content demand forecasts to provide significantly better cache performance metrics. In addition to proposing MPU, two additional caching algorithms, LFU-W and LRU-W are also used to demonstrate the benefits achievable by leveraging demand forecasts, which, in spite of performing worse than MPU, perform much better than traditional caching algorithms.

The use of MPU enables significant cache costs savings, for a fixed target hit-ratio, or much better caching performance when run using identical storage resources.

To validate the proposed caching algorithm in experimental scenarios, this work is further developed and tested in a content-aware delivery solution on the next section.

4.4 Content-Aware Over-The-Top Delivery of Catch-up TV Services

Considering the reference CDN architectures in Section 2.3, the characterization of Catch-up TV on Chapter 3, the QoE evaluation framework on Section 4.1, the exploration of demand forecasting mechanisms on Section 4.2, and the caching algorithms that leverage forecasts to improve their performance on Section 4.3, this section proposes a new delivery architecture that maximizes the performance of multimedia OTT CDNs as a whole, through content-aware mechanisms.

All the previous research works and literature reviews are put together to form a cohesive end-to-end delivery architecture that is capable of addressing the needs of modern users with high-QoE while ensuring an efficient operation.

In spite of being tailored towards Catch-up TV, which is presented as a use-case, the architecture herein described is generalizable to other services and content types. The complete paper is included in *Appendix G*.

4.4.1 Introduction & Motivation

The application of "standard" CDNs to multimedia streaming delivery and, in particular, to Catch-up TV delivery is far from optimal, as this type of content exhibits a dynamic demand behavior that is not properly accommodated by traditional CDN replica servers [52]. A CDN should not be agnostic of its content so that better performance levels are achieved, hence the need for *content-aware* CDNs. *Content-awareness* refers to the adaptation of data storage, processing or transmission methods according to characteristics of the content being delivered, and is highly dependent on the systems' ability to extract meaningful information from it.

Considering these issues, and the fact that the overall performance of CDNs is highly dependent on the efficient usage of the available servers, measured in computational, memory, and network requirements, this section proposes, details, and evaluates a novel content-aware caching architecture capable of leveraging content-aware demand forecasts produced by a predictive machine learning model to provide dynamic resource allocation capabilities while simultaneously improving caching decisions considering specific characteristics of the content requested, i.e. in a *content-aware* manner.

4.4.2 Scientific Contributions

The work presented in this section brings together a diverse but complementary set of scientific contributions that consolidate the previously discussed research, namely:

- Proposal of a novel content-aware OTT delivery architecture with a detailed discussion and modeling of its building blocks', features and responsibilities;
- Proposal of a prediction algorithm based on machine learning to forecast Catch-up TV programs' requests;

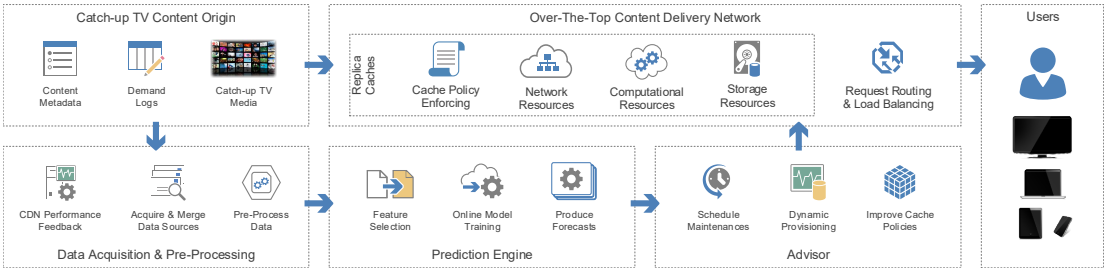


Figure 4.16: Proposed Content-Aware Over-The-Top Catch-up TV Delivery Architecture.

- Proposal of an advisor algorithm that decides on the distributed caching configuration to optimize CDN performance with cache size minimization;
- Experimental implementation of the proposed architecture targeting a Catch-up TV delivery use-case;
- Performance validation of the content-aware delivery architecture using requests logs from a production Catch-up TV service, considering key QoS metrics and QoE estimations.

4.4.3 Proposed Content-Aware Over-The-Top Delivery Architecture

Having considered the potential benefits of content-aware approaches to improve delivery systems, on Section 2.3.6, along with key Catch-up TV characteristics that are essential to the design of an optimized OTT delivery solution on Section 2.2.3, this section proposes a new architecture that maximizes the performance of Catch-up TV OTT CDNs through content-aware mechanisms.

Figure 4.16 exhibits the envisioned global architecture along with its main components. A macro overview of the proposed architecture presents 6 different functional blocks. The *Catch-up TV Content Origin* is responsible for holding the complete set of Catch-up TV content, the associated metadata, and user request logs. Next, the *Over-The-Top Content Delivery Network* block represents the actual system responsible for the efficient and high-QoE delivery of Catch-up TV contents, through the use of replica caches, to the *Users*, which are the final consumers of the Catch-up TV service.

The remaining 3 functional blocks are responsible for managing and ensuring the optimal operation of the *Over-The-Top Content Delivery Network*. The purpose of the *Data Acquisition & Pre-Processing*, *Prediction Engine*, and *Cache Advisor* blocks is to create and distribute dynamic provisioning and caching policies to be used by the *Over-The-Top Content Delivery Network*. By working as a complement, and in parallel to the main content delivery flow, this architecture enables non-disruptive improvements to current CDN solutions. The detailed responsibilities of each individual element and sub-elements are provided in the ensuing subsections.

Catch-up TV Content Origin

This component, commonly known as *Origin*, aggregates three main responsibilities:

- *Content Metadata* contains information that is associated with each media element. In Catch-up TV services, content metadata includes EPG information, such as the original program airing date, broadcast station, program title, episode number, series identifier, and duration, to name a few;
- *Demand Logs* provide traceability and accountability by recording who requested what and when, therefore providing timestamped records that associate user requests to Catch-up TV programs;
- *Catch-up TV Media* is the actual media vault responsible for holding the encoded media elements, usually also encrypted, ready to be delivered to the end-users.

In practice, the *Origin* may also have to interface with external Business Support Systems (BSSs) and Operations Support Systems (OSSs), but these are its main responsibilities.

Over-The-Top Content Delivery Network

This block is the optimization target on the overall content-aware architecture. It is often composed of multiple servers, called the replica, surrogate, or cache servers, and is responsible for delivering Catch-up TV content from the *Origin* to the end-users. The replica servers are interconnected and store content copies to reduce the load on *Origin* servers and network interconnect, while also increasing the services' QoE.

They may be characterized by their *Computational*, *Storage*, and *Network* resources, which should be adequately dimensioned taking into the consideration the services' QoE vs. OPEX/CAPEX trade-off. Given that it is often not economically viable to fully replicate the *Origin's* content, replica servers must employ caching strategies to carefully select what to keep in storage and what to discard, thus *Cache Policy Enforcing* is a key function of replica servers.

In addition to the replica servers, *Request Routing & Load Balancing* systems are also required to properly direct users and traffic to the most suitable replica servers.

Users

The *Users* element represents the services' consumers. They may be geographically dispersed and use any Internet-connected device to access Catch-up TV content on-demand. It is important to properly model the users' demand profiles to adequately tune and dimension the CDNs' resources, i.e. network, storage, and computing.

Data Acquisition & Pre-Processing

The *Acquire & Merge Data Sources* element is in charge of interfacing with data-sources that contain relevant information regarding the content being cached and merging it into meaningful representations.

For Catch-up TV, suitable data-sources include the EPG, *Content Metadata*, analytics events providing information regarding users' requests and preferences, i.e. *Demand Logs*, as well as *CDN Performance Feedback* metrics.

A meaningful data representation maps a set of user requests to a specific TV program, accompanied by its metadata — such as the original airing date, TV station, etc. — along with prior CDN performance metrics for that particular program.

The past performance metrics create a feedback loop that aids the accuracy of future predictions by providing information regarding past prediction errors.

After the initial data acquisition and merging process, *Pre-Processing* is applied in order to compensate for discrepancies caused by the predictors' different scales, standard deviations, and average values. These discrepancies in scale and statistical properties often impair the numerical stability and bias of learning algorithms, potentially favoring some predictors over others, not because of their real importance but because of their different scales; therefore, it is important to scale, center, and correct the skewness of each predictor before making the data available to the *Prediction Engine*.

Prediction Engine

The prediction engine is key in this content-aware approach, and it is where the learning and forecasting cores of the content-aware caching solution are implemented. A data-driven simulation study on how to derive a forecasting model from the available Catch-up TV dataset is explored on Section 4.2.

The prediction engine's responsibility is to gather inputs from the *Data Acquisition & Pre-Processing* component and to generate accurate predictions regarding future Catch-up TV programs' requests that influence the CDN's configuration and performance. Depending on the available data and topology, the module may be required to predict consumer demand per PoP.

A mathematical description is hereby presented to clarify the operations performed by *Prediction Engine*'s components. P represents the set of p unique Catch-up programs, S comprises the set of s available predictors that describe each log entry, – containing program, user, and CDN performance data, as described in the previous sections –, and L is matrix of log entries, with m rows, and $|S|$ columns.

We define t as a timestamp variable, measured since the epoch (1970-01-01 00:00:00 UTC), in hours – Equation 4.4. Empirical findings and prior data analysis [28, 26] indicate that 60 minutes time slots represent an adequate compromise between time precision and computing requirements, even though specific scenarios may require a better time resolution.

$$t = \left\lfloor \frac{\Delta t_{epoch}}{3600} \right\rfloor \tag{4.4}$$

Feature Selection To generate demand forecasts per program, this block ingests data from the *Data Acquisition & Pre-Processing* component, leveraging supervised and un-

supervised techniques to perform an initial selection of predictors. Supervised methods rely on previous data and known outcomes, while unsupervised approached do not.

In this work, a supervised *filter* method is employed based on *ensemble* selection, implemented in R's *fscaret* package [308]. Unsupervised selection is performed through Near-Zero Variance (NZV) and cross-correlation analyses [257]. The feature selection process takes into account the fact that a Catch-up TV program must be unequivocally identifiable using a minimum set of predictors. Equation 4.5 illustrates the log data matrix L, with m log entries and $|S|$ predictors.

$$L = \begin{pmatrix} l_{11} & \dots & l_{1n} \\ \vdots & \ddots & \vdots \\ l_{m1} & \dots & l_{mn} \end{pmatrix} = (l_{in}) \in \mathbb{R}^{m \times n} : n = |S| \tag{4.5}$$

The filtering process selects a subset $S' \subset S$ of predictors as a result of the individual techniques. S' is presented on Equation 4.6, where S_S represents the set of filtered predictors using supervised methods, and S_U using unsupervised methods.

$$S' = S_S \cup S_U : S' \subset S \tag{4.6}$$

As a result of the filtering process, the final set of log data to be used in the subsequent steps relies only on the filtered S' predictors, so that matrix L' contains the same number of log entries as L, but with $|S'|$ predictors only – Equation 4.7.

$$L' = (l_{mn}) \in \mathbb{R}^{m \times n} : n = |S'| \tag{4.7}$$

Online Model Training After defining the forecasting constraints and time-granularity decisions, the *Online Training Model* block leverages the selected predictors S' to retrain a Random Forest (RF) machine learning algorithm [303], using the filtered matrix L'. The retraining function is illustrated in Equation 4.8, denoted by $T()$, whose parameters are M_t – the latest forecasting model at time t – and L'_{t+1} – the newly filtered data matrix. As a result, the training function generates a new forecasting model M_{t+1}, which will be used on the forecasting step. Even though other regressive algorithms might be employed, as long as online model training is supported, our previous findings suggest that RFs have good predictive capabilities for this particular use-case.

$$T(M_t, L'_{t+1}) = M_{t+1} \tag{4.8}$$

On the following step, the updated forecasting model M_{t+1} will be used to *Produce Forecasts* for each program expected to air in the period under analysis.

Forecasting The final process is to *Produce Forecasts* for each program p expected to air in the period under analysis, e.g. $t + 1$, with detailed demand predictions for each program. Equation 4.9 presents a forecasting function $F()$ taking as input the

latest forecasting model M_t, the target forecasting time slot $t+1$ and program p, and outputting a program demand estimate for the desired period.

$$F(M_t, t+1, p) = D_{p(t+1)} \quad \forall p \in P \tag{4.9}$$

As a result of the forecasting function, a full estimate on the upcoming program demand is achieved, and the generated forecasts are pushed to the *Advisor*, whose purpose is to manage and distribute configurations to replica CDN nodes.

Advisor

Demand forecasts produced by the *Prediction Engine*, D_{pt}, are leveraged by the *Advisor* in three different manners.

First, from an operational perspective, knowing when users' demand is the lowest helps to optimally *Schedule Maintenance*, such as running software updates, file-system checks, or other operations that would be undesirable when the systems are heavily loaded, in order to prevent a negative QoE impact.

A scheduling algorithm example is described by Equation 4.10, which picks the best maintenance time slot according to the expected total demand, within a given time-window W defined for a set of possible time slots $\{t, t+1, \ldots\} \in W$.

By providing additional constraints, such as PoP location, this formulation may be trivially expanded to produce maintenance schedules specific to individual PoPs.

$$\text{minimize} \quad \sum_{p \in P} D_{pt} \qquad \text{subject to} \quad t \in W \tag{4.10}$$

Second, from a cost optimization perspective, accurate forecasts enable aggressive dynamic resource provisioning policies, where significant power, computational, bandwidth, and storage savings are possible without compromising the services' performance and users' QoE. From the D_{pt} forecasts it is possible to predict the amount of required storage and bandwidth, given that they depend directly on the characteristics of program p and its demand at time t.

Finally, due to the detailed knowledge on future demand, the *Advisor* is also responsible for acting as a coordination agent for distributed caching configurations with the purpose of optimizing replicas' caches. As shown on Section 4.3 in a simulation environment, priority maps derived from demand forecasts have the potential to significantly improve caching performance.

4.4.4 Experimental Evaluation

This section describes the implementation and testing procedures used to validate the proposed architecture relying on readily available solutions for OTT CDNs.

Training and Testing Data

The dataset's quality is critical for the performance of any forecasting algorithm. In this work, the dataset described on Section 3.3 is used.

The request logs are split into 2 different groups, according to their purpose. The first group, reserved for *training*, is comprised by the initial 23 days of logs, while the remaining 7 days are held up and used for performance assessment purposes. Considering the previously established time slot granularity of 1 hour and the testing period, a total of $7 * 24 = 168$ demand forecasts are computed.

Catch-up TV Content Origin

Demand Logs & Content Metadata The training dataset contains both the demand logs and the associated content metadata; therefore, these components get their information from the same data source.

Catch-up TV Media For the media vault, a Microsoft IIS server is set up with Smooth Streaming [37] content to mimic the expected OTT scenarios leveraging adaptive bitrate encoded content. For practical reasons, regardless of the Catch-up TV program requested, the same video content is always provided. When crafting the content request URL, the query strings are modified to ensure that the CDN treats each program independently.

Over-The-Top Content Delivery Network

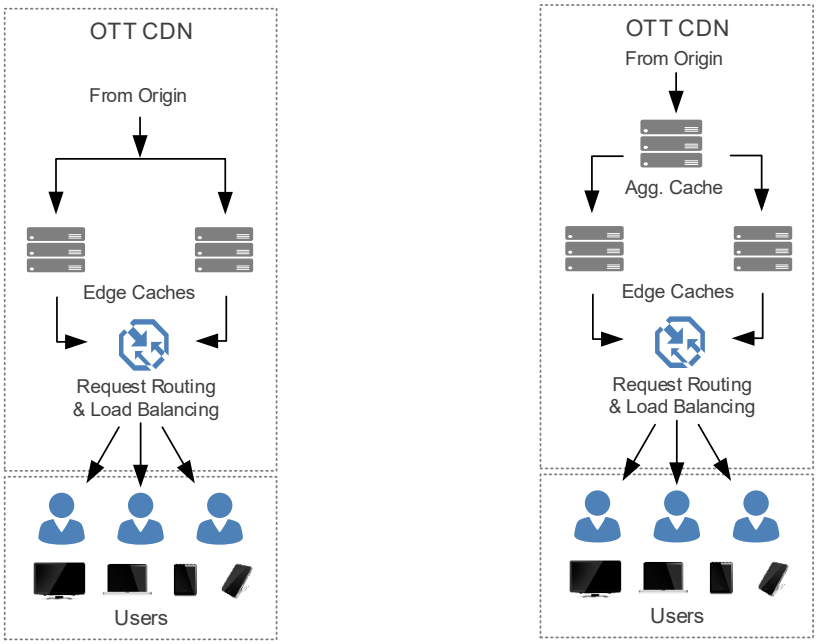
CDN architectures are suitable to *proxy-cache* deployments; therefore, the experimental validation focuses on replica cache solutions with 1 and 2-tier caching layers, i.e. with and without *Aggregation Caches*, as depicted in figures 4.17(a) and 4.17(b).

Even though most common proxy-cache solutions, such as Nginx [77], and Squid [79] are open-source and modifiable, a choice was made to use ATS [75] for implementing the custom *Edge Caches* and *Aggregation Caches*. The reason for this choice was of practical nature, as this project's code is well documented and easy to extend.

The *Request Routing & Load Balancing* tasks are handled by HAProxy [314], which uses a Round-Robin strategy to randomly distribute requests within the Edge Caches.

Users

Catch-up TV users are simulated through Python scripts performing HTTP requests to the load balancer. In order to ensure an accurate reproduction of real scenarios, the requests are performed sequentially, according to their original order in the previously described 7 days training request logs. The ordered and predictable nature of users' requests ensures that different test runs produce comparable results.



(a) 1-Tier Caching. (b) 2-Tier Caching.

Figure 4.17: Experimental Replica Cache Architectures.

Testbed Description

Tests are run in a virtualized environment (VMware ESXi 5.5.0), using an HP ProLiant DL160 Gen9 server with 2 x Intel E5-2640v3 CPUs (32 cores) and 32GB of RAM. The detailed resource reservations per component are presented on Table 4.1. An additional identical server is connected to the management network and is used for the *Data Aquisition and Pre-Processing*, *Prediction Engine*, and *Advisor* tasks.

Caching Algorithms

Caching algorithms play a crucial role on a CDN's overall performance. To leverage the content-aware demand forecasts, it is important to use algorithms that are able to benefit from that additional knowledge. Three caching algorithms, MPU, LRU-W and LFU-W – presented on Section 4.3 –, are implemented to take advantage of the demand predictions, in addition to standard LFU, LRU, and FIFO, which are also implemented in ATS and benchmarked. The custom LRU-W and LFU-W use the demand predictions to weight the importance of cache items. Even though other caching algorithms exist, most are either variations or combinations of the aforementioned algorithms. The algorithms' implementations are kept as similar as possible, and their behavior is cross-checked with simulations in R.

	Load Balancer	Edge Caches	Aggregation Cache	Users	Origin Server
# Instances	1	2	1	1	1
Software	HAProxy 1.5.11	ATS 5.3.0	ATS 5.3.0	Python Script	IIS 8.5
CPUs	4	6	6	6	4
RAM	4GB	6GB	6GB	6GB	4GB
NICs	2 x 10GbE (Data + Management) with 9000 MTU				
OS	Ubuntu 14.04.1 LTS x64				Win. Server 2012 R2 x64

Table 4.1: Virtual Machines (VMs)' Technical Details per Instance.

Cache Sizing

To explore the effect of different cache sizes in the experimental tests conducted, the caches are sized as fractions of the total number of unique available programs. Therefore, a cache size of 100% corresponds to a cache with the ability to hold the entire content catalog available on the 7 days testing window. Each program is assumed to require 1 storage unit. Given its purpose of reducing the load on the Origin server and serving as an intermediate cache, the Aggregation Caches are always sized with twice the storage of the Edge Caches.

Performance Metrics

To understand the improvements provided by the envisioned content-aware OTT CDN solution, it is necessary to define the metrics by which the delivery infrastructure is evaluated. These metrics are assessed according to how they vary along two vectors: cache size and time. By exploring performance variations with cache size, it is possible to determine the cost-benefit trade-off of improving caches' sizes, while the performance variation with time is essential in Catch-up TV services with dynamic content popularity that may impact the metrics under evaluation. In order to perform the time-varying analyses the cache sizes are set at 1% of the total corpus.

Cache Hit-Ratio This metric summarizes the ratio between the number of cache hits when compared to the number of cache requests, and is an indicator of how good the caching algorithm is on guessing programs that will be requested in the near-future.

Backend Traffic A key cost factor in content distribution is the backend traffic requirements within the CDN infrastructure — including the *Origin* —, before delivering the content to the users. As the purpose of replica caches is to reduce the load on backend servers, a low metric is indicative of good performance.

Request latency The time required to service a request is an important performance indicator that must be carefully monitored to ensure that there is no impact on the user experience, as high request latency may indicate high server or network load, which leads to queued or dropped requests.

QoE MOS From a user's perspective, what ultimately matters is the overall service QoE, measured through a MOS. As defined by ITU-T [221], QoE is: *"The overall acceptability of an application or service, as perceived subjectively by the end-user"*. Due to its subjective nature, QoE evaluations vary significantly between different users; however, objective QoE evaluation frameworks exist that provide an estimate on the expected MOS of a given service. A previously developed Smooth Streaming QoE estimation probe, detailed on Section 4.1, is used to provide an objective MOS estimate.

4.4.5 Main Results

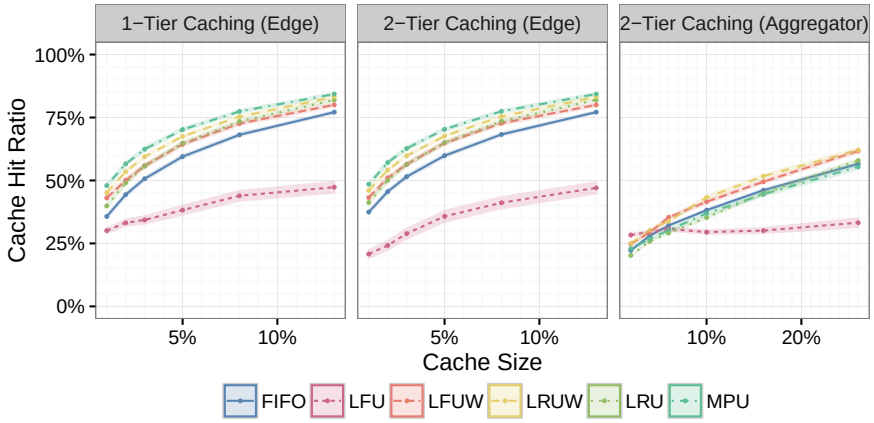
The validation of the proposed architecture is essential to draw conclusions regarding the feasibility and actual performance of the solution. As this architecture is suited towards proxy-cache deployments, the experimental validation focuses on solutions with 1 and 2-tier caching layers, i.e. with and without *Aggregation Caches*.

When pertinent, the results are presented in a normalized fashion, ranging from 0% to 100%, to facilitate a graphical analysis, and the 95% CI is shown on the average values' curve and data points.

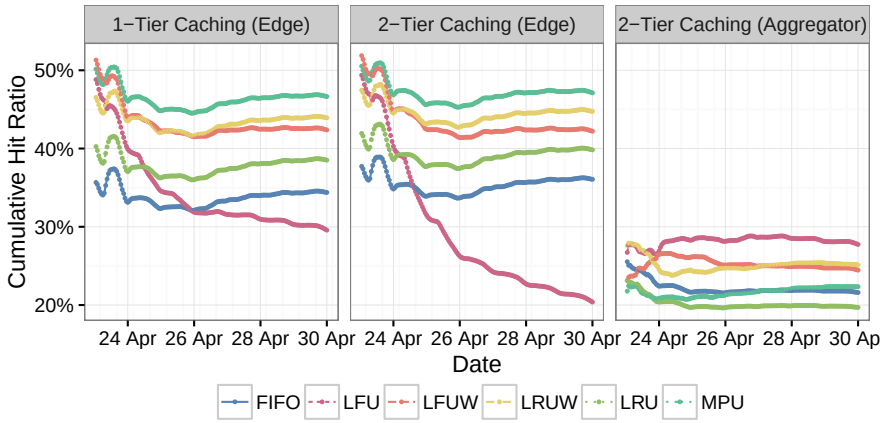
Cache Hit-Ratio Variation with Cache Size

The first analysis explores the impact of different cache sizes in the caches' hit-ratios. The results are presented in Figure 4.18(a).

It is evident, from the results, that the usage of demand forecasts as inputs to caching algorithms, specifically to MPU, LFU-W and LRU-W, is helpful in significantly improving the servers' caching performance, in both 1-tier and 2-tier caching scenarios – particularly for smaller cache sizes (1 to 3%). As expected, the cache hit-ratios at the edges in both scenarios are similar, while the cache hit-ratios at the aggregation cache are significantly lower when compared to that of the edges, despite its larger cache size. The cause for this behavior is related to the fact that highly popular items stay cached at the edges and are rarely requested from the aggregation cache, which instead ends up caching, and generating cache hits, for items that fall out of the edges' caches. It is interesting to observe that best caching policy at the edges, MPU, is not necessarily also the best one to be implemented at the larger aggregation cache. Instead, LRU-W, due to its aging policies, takes the lead in aggregation caches. This effect is due to the very different traffic patterns that each cache tier observe. While the edges are directly serving clients, the aggregation caches' main purpose is to compensate the edges' misses.



(a) Variation with Cache Size.



(b) Variation with Time.

Figure 4.18: Cache Hit-Ratio Results.

Cache Hit-Ratio Variation with Time

The results of Figure 4.18(b) demonstrate how caching performance varies with time. It is possible to observe that, as time progresses, some algorithms adapt better than others to content requests. As with the previous analysis, for edge caches, MPU provides the best results, closely followed by LRU-W and LFU-W, proving once more that adding content-awareness to CDNs has the potential to significantly improve their performance. In spite of excellent LFU results for the early hours of day 24, its hit-ratios' performance progressively diminishes with time, which might be explained by the effect of "cache pollution", whereby items that were initially highly popular, but lose relevance, prevent other newer items from populating the caches; hence, leading to low hit-ratios. The small increase in hit-ratios on all algorithms in day 24 is believed to be due to accentuated users' demand for popular content in some times of the day, i.e. a result of the *superstar* effect which also happens in the remaining days, albeit at a smaller scale.

As for the aggregation cache, the hit-ratios are much lower than those at the edges, with LFU taking the lead, closely followed by LFU-W and LRU-W.

Overall, considering the edges and aggregation cache, MPU, LRU-W, or LFU-W yield the best performance.

Backend Traffic Variation with Cache Size

The volume of backend data transfers is a metric that impacts the scalability and cost of CDNs. On the one hand network traffic to/from origin servers is usually expensive when compared to traffic between the clients and the edge caches, while on the other hand, origin servers are typically not dimensioned to be able to cope with direct demand from all users and require the fan-out capacity provided by edge and aggregation caches.

The edge caches' backend traffic is summed, thus, in 1-tier caching it represents the total amount of traffic between both edge caches and the origin server, while on the 2-tier scenario it shows the total traffic between the edge caches and the aggregation cache.

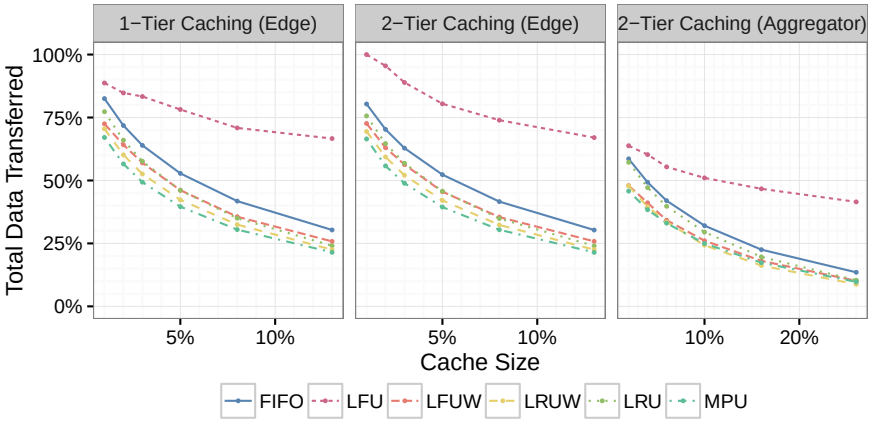
To explore how this metric varies with the cache size, Figure 4.19(a) presents detailed information regarding the total backend traffic of each component in both 1-tier and 2-tier scenarios which, as expected, are almost identical.

It is possible to observe that the inclusion of an aggregation cache reduces the traffic to the origin server in approximately 35% regardless of the cache algorithm chosen, in addition to aiding in edge caches' rebuilds in the event of failures.

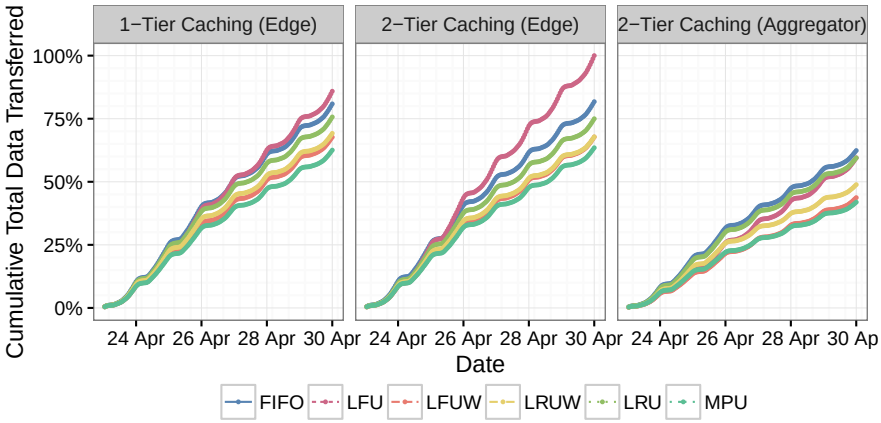
The outcomes are a reflection of the cache hit-ratios' results presented in the previous section, and demonstrate that higher cache hit-ratios lead to a reduction in the backend data transfers, as expected.

Backend Traffic Variation with Time

As a complement to the previous study, this analysis focuses on the evolution of cumulative backend data transfers with time, which are expected to evolve inversely proportionally to the hit-ratios of each solution.



(a) Variation with Cache Size.



(b) Variation with Time.

Figure 4.19: Backend Traffic Results.

Figure 4.19(b) shows that, while every caching algorithm starts with approximately the same amount of data transferred, as time progresses, they quickly diverge.

The observable periodic pattern reflects the varying content demand at the different times of day. Periods with reduced demand – late night and early mornings – show up as almost horizontal segments, while the periods with high demand are responsible for the sharp traffic increases.

At the edge caches of 1-tier and 2-tier scenarios, by the end of 7^{th} day, the best performing caching algorithm, MPU, transfers 28 to 37% less data than the worst performing algorithm, LFU. The next best performing algorithms, LRU-W and LFU-W, only transfer $\sim$ 7% more data than MPU, while LRU and FIFO require, respectively, $\sim$ 15% and $\sim$ 23% more backend data than MPU.

Analyzing the aggregation cache of the 2-tier scenario, the first observation is that the total data transfers performed to the origin are significantly lower than those of the edge caches, while the performance differences between the distinct caching algorithms are also significant, with MPU and LFU-W taking the lead, closely followed by LRU-W. The remaining traditional caching algorithms, LFU, LRU, and FIFO impose a much larger strain on the origin.

Request Latency Variation with Cache Size

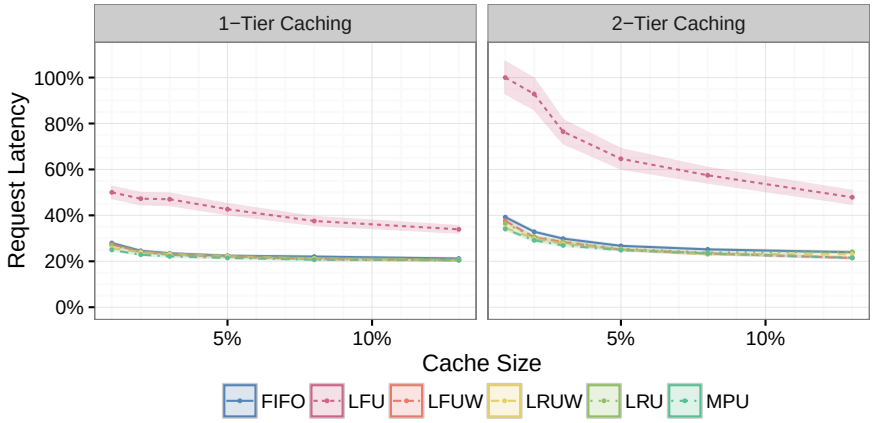
The variation of request latency with cache size is presented in Figure 4.20(a), where it is clearly observable that, except for LFU and, at a smaller scale, LFU-W, all caching algorithms present similar request latency metrics, that fall within each other's confidence intervals. Nevertheless, it is possible to perceive a slight overall reduction on the request latency for all caching algorithms as the cache sizes increase. This effect may be due to better cache hit-ratios for large cache sizes. It is worth to point out that in the experimental scenario, using VMs, no significant network delays exist between the different VM instances.

Request Latency Variation with Time

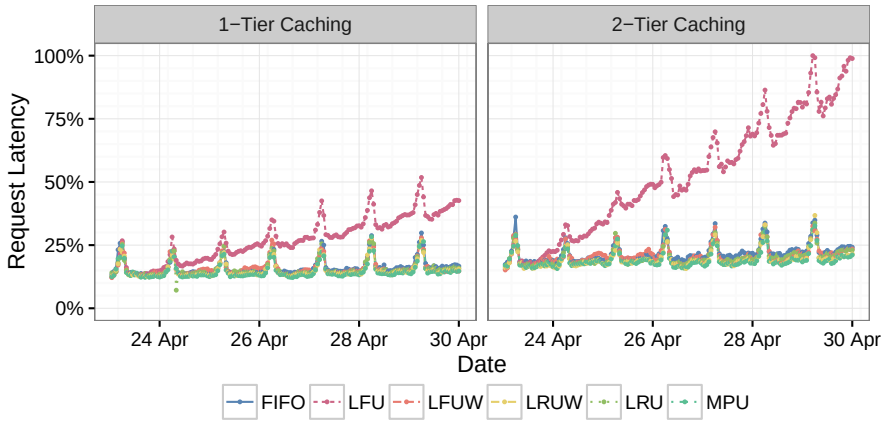
The results of Figure 4.20(b), which explore the variation of request latency with time, reveal similar conclusions to those taken on the previous analysis. LFU is clearly the worst-performing caching algorithm, with a run-away latency metric that points to request-queuing and server overloading. LFU-W fares worse than its remaining counterparts, which exhibit a consistent and similar request latency performance throughout the evaluation period.

QoE Variation with Cache Size

Figure 4.21(a) reveals that measurable benefits are achievable by increasing the caches' size, which is a direct impact of better cache hit-ratios, in both 1-tier and 2-tier scenarios. MPU and LRU-W provide the most significant MOS improvements over



(a) Variation with Cache Size.



(b) Variation with Time.

Figure 4.20: Request Latency.

traditional caching algorithms, closely followed by LFU-W and LRU. FIFO and LFU provide the worst MOS for every considered cache size.

By adding an aggregation cache, in the 2-tier scenario, it is possible to observe that a slight MOS improvement is achievable over 1-tier scenarios for all caching algorithms.

These results demonstrate that the proposed content-aware solution is capable of boosting the performance of caching algorithms in 1-tier and 2-tier scenarios, from a technical perspective – higher hit-ratios, reduced data transfers and request latency – and from a user's perspective, in the form of MOS enhancement.

QoE Variation with Time

The results of Figure 4.21(b) complement those of Figure 4.21(a), by showing that, with the exception of LFU-W and LFU, the performance of the remaining caching algorithms remains consistent throughout the period under analysis for both 1-tier and 2-tier scenarios. As with the previous results, MPU and LRU-W clearly dominate this evaluation and provide a significantly improved MOS when compared to the other caching algorithms.

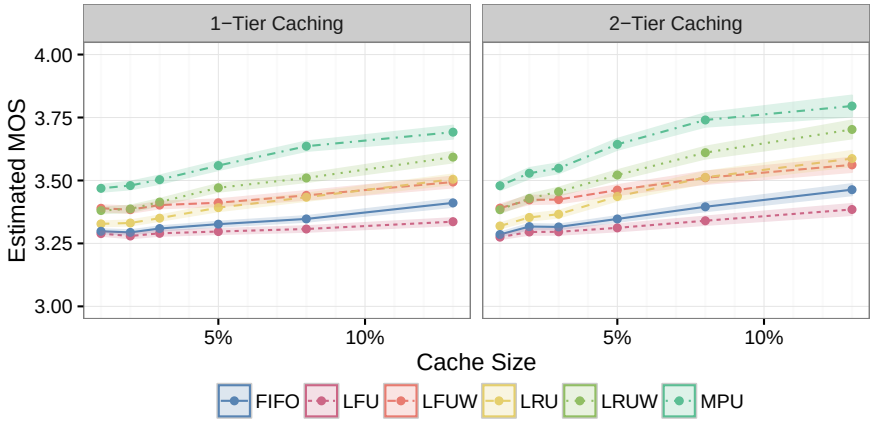
Once again, LFU and, to some extent, LFU-W, appear to suffer with the issue of "cache pollution", which is reflected on their performance degradation with time.

4.4.6 Conclusion

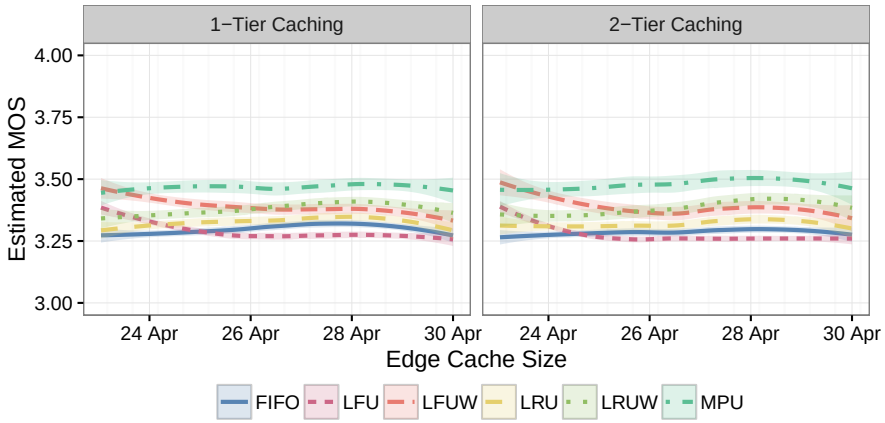
The migration of managed Catch-up TV services to OTT poses several challenges that must be addressed by next-generation delivery solutions, which must improve the services' QoE, while reducing their CAPEX and OPEX.

A content-aware architecture is proposed and thoroughly detailed that leverages machine-learning and data-mining techniques to forecast content demand, improve caching policies and facilitate autonomic resource management.

The proof-of-concept experimental implementation of the content-aware OTT delivery architecture is evaluated under realistic conditions, using request logs from a popular production Catch-up TV service to validate its design. The experimental results show that the enhanced architecture, with caching algorithms capable of taking advantage of content knowledge – MPU, LRU-W, and LFU-W –, significantly outperform reference implementations in terms of cache hit-ratios, bandwidth savings, request latency, and users' QoE, opening the door for future, smarter, and even more efficient delivery solutions capable of leveraging content characteristics to continuously and dynamically improve themselves.



(a) Variation with Cache Size.



(b) Variation with Time.

Figure 4.21: Estimated MOS.

Chapter 5

Conclusions and Future Work

This Chapter provides the main conclusions and impacts on the addressed research areas, a summary of the main achieved results and contributions towards the Thesis' goals, and a discussion on possible future directions and open challenges that were not addressed but are relevant for upcoming research efforts.

5.1 Final Remarks / General Conclusion

This Thesis' begins by introducing OTT multimedia delivery solutions, why they are important, and what is the expected role of OTT multimedia delivery in the future Internet. An overview is provided on the research approach, its main objectives, and why Catch-up TV is chosen as an optimization use-case.

The reference end-to-end multimedia delivery pipeline provides a holistic view of the components contributing to the performance of OTT multimedia services. Its key research areas are identified and surveyed, starting with CDNs, proceeding with streaming technologies and protocols, QoE, caching issues, and data mining challenges. The state-of-the-art survey provides solid scientific grounds for identifying the key open issues on each area, and serves as a guideline for shaping the main research goals, which are broken down into smaller and more focused targets to be addressed.

Having understood the main scientific gaps in the current state-of-the-art, an initial Catch-up TV service characterization, leveraging a large production dataset, demonstrates the service's relevance, while fostering a detailed understanding on users' viewership dynamics and associated challenges.

To improve the performance of content delivery solutions, several new models, approaches, and enhancements are presented, starting with a QoE estimation tool for novel HAS scenarios, continuing with research work on demand forecasting through machine-learning, and improved caching algorithms, which are shown to outperform competing approaches. The individual contributions fit together in an envisioned content-aware OTT delivery architecture that is able to continuously and seamlessly adapt to its content, enabling improved cache hit-ratios and dynamic resource provisioning solutions, while enhancing the overall user experience.

5.2 Contributions and Results

Modeling and Characterization of Catch-up TV Services

A thorough understanding of a problem's domain is essential before any optimization work, as it frames, guides and shapes improvement opportunities. To that end, research is conducted with the purpose of assessing the relevance of timeshift services, and Catch-up TV in particular, from a worldwide perspective. These revolutionary services are essential to fight churn and cord-cutting on modern viewers, and change the way how broadcasters measure their ratings. The broad scope of this evaluation makes it clear that Catch-up TV services are widespread and massively popular, to the point that they are actually shaping and empowering consumers' behaviors.

In addition to framing the service's relevance, a detailed statistical characterization, unique in its scale, depth and detail, is performed from a large dataset acquired from a production service, which enables an exclusive insight on what, when, and how users watch Catch-up TV content and provides statistical summary tables that may be used as reference by other researchers in the field.

The user characterization performed concludes that most clients access the service through a single STB; they are very active throughout the whole day – especially on weekends; have a marked preference for content with *General*, *Kids*, and *Movies and Series* genres; and favor serialized content over one-off programs.

The most popular programs aired during prime-time and exhibit the *superstar* effect, i.e. Catch-up TV reinforces their popularity instead of promoting a *long-tail* model. A significant portion (∼40%) of the content catalog is never accessed.

A usage analysis reveals that users take, on average, 2m:32s to find the program they want, indicating a potential optimization opportunity in terms of user experience and program discovery. Moreover, Catch-up TV programs get 75% of their total requests within the first 3 days of airing, suggesting that their relevance is highly time-sensitive.

From a content delivery perspective, the service optimization analyses revealed large differences between peak and off-peak bandwidth demand, which is problematic due to the underutilization of network resources, which needs to be dimensioned to approximately two times the average streaming bandwidth to avoid network bottlenecks.

These conclusions point to significant service improvement possibilities that can and should be used on CDNs to provide a better QoE to users, while simultaneously reducing Pay-TV operators costs.

The full contributions are presented in the following publications:

- *Time-shift services: a taxonomy and techno-business impacts of Catch-up TV* – Appendix A [25];
- *Survey of Catch-up TV and Other Time-Shift Services: A Comprehensive Analysis and Taxonomy of Linear and Nonlinear Television* – Appendix B [27];
- *Catch-up TV Analytics: Statistical Characterization and Consumption Patterns Identification on a Production Service* – Appendix C [28].

Content-Aware Over-The-Top Delivery of Catch-up TV Services

The modeling and characterization works on Catch-up TV services hint at several optimization opportunities to be had on the content delivery pipeline.

To be able to properly assess QoE on next-generation OTT multimedia delivery systems relying on HAS, adequate QoE modeling and evaluation tools are required. The lack of suitable tools and models on the existing literature motivates the proposal and development of a HAS-tailored MOS estimation model capable of taking into account parameters such as the devices' platforms, display sizes, computational capabilities, and other dynamic characteristics such as perceived link bitrate, actual decoded Fps, chunk quality, and a human-like memory effect that takes into consideration the impact of buffering events and number of quality switches to produce MOS estimates that are shown to be in line with actual MOS classifications from users.

Considering the large Catch-up TV dataset at our disposal, and the potential benefits achievable by creating accurate demand forecasts, to which the previous statistical characterization pointed, this Thesis contributes with a thorough and detailed approach for creating forecasting models using a set of predictive machine learning algorithms. The evaluation of the predictive capabilities of the different algorithms is explored not only from a purely statistical perspective, but also from a service optimization perspective, through which its usefulness is shown, particularly to achieve savings on storage and bandwidth, essential in highly efficient CDNs.

To showcase other benefits brought forward by suitable demand forecasting algorithms, the Thesis also contributes with a novel caching algorithm, MPU, that is able to leverage content demand forecasts to provide significantly better cache performance metrics. The results show that the use of MPU enables either significant cache costs savings, for a fixed target hit-ratio, or much better caching performance when run using identical storage resources.

All the previously contributions fit together as individual pieces of a larger content-aware OTT delivery architecture, which addresses the complete content pipeline to provide an integrated optimized solution capable of fully leveraging the information at its disposal to create a high-performance delivery solution. To validate the applicability and feasibility of these contributions, a thorough experimental validation is conducted using widely deployed open-source solutions. In spite of its application to the Catch-up TV use-case, contemplated in this Thesis, the experimentally validated envisioned architecture may also be applicable to other content delivery scenarios.

The full contributions are presented in the following publications:

- *QoE Assessment of HTTP Adaptive Video Streaming* – Appendix D [31];
- *Catch-up TV Forecasting : Enabling Next-Generation Over-The-Top Multimedia TV Services* – Appendix E [29];
- *Over-The-Top Catch-up TV Content-Aware Caching* – Appendix F [28];
- *Content-Aware Over-The-Top Delivery of Catch-up TV Services* – Appendix G [30].

5.3 Future Work

Even though this Thesis addressed several research challenges on the end-to-end content delivery pipeline of OTT CDNs, many other opportunities exist, in terms of understanding how these novel services are shaping users' behaviors and habits, and with respect to additional improvements that have the potential to take the envisioned content-aware delivery architecture to higher grounds.

From a social and behavioral perspective, open challenges include an exhaustive comparison of the previous research works on IPTV content demand characterization to understand if, and how, users' behaviors are changing. The rising popularity of nonlinear services points to changes in the modern TV watching paradigm, with impacts on users' lives and social interactions that should be considered.

As a result from the Catch-up TV service characterization, it is possible to observe that a significant portion of the available content ($\sim$40%) is never requested, leading to unnecessary usage of scarce storage resources. Future work will include studies to explore how this unimportant content might be identified to either prevent its recording, or to delete it as soon as possible.

The developed demand forecasting models should also be improved to take into consideration that different geographical regions may exhibit divergent demand patterns that must be accounted for, with the purpose of maximizing forecasting accuracy.

Regarding the possibilities enabled by content-aware delivery systems relying on accurate demand forecasting engines, a very appealing proposition is that of being able to prefetch content into users' devices before they request it, to significantly reduce the peak bandwidth usage, which was shown to be as high as 10 times the minimum bandwidth requirements. By "flattening" the bandwidth curve, the likelihood of network-related issues on peak hours is reduced, the overall service quality increases, and smaller investments are required on network capacity that is most of the times unused.

The developed caching algorithm is shown to exhibit an excellent performance in content-aware situations; however, it is also important to understand how it behaves on other scenarios, such as when faced with unknown content, in order to be able to improve its robustness and add new features such as scan-resistance. Additional improvements may also encompass the development of cooperative distributed caching strategies, and studying its behavior with other HAS implementations, such as MPEG-DASH or HLS.

Even though this Thesis' work is centered around the Catch-up TV use case, an important future research direction is that of expanding this work to contemplate other scenarios, such as commercial VoD catalogs, or user-generated content. The validation and generalization of the content-aware OTT CDN is essential to assert the importance and benefits to be had by the envisioned solution.

Bibliography

[1] F. Venturini, "Bringing TV to Life, Issue II: The race to dominate the future of TV." Accenture, 2011, no. II, Accessed: 09-2015. [Online]. Available: https://www.accenture.com

[2] C. Cai, J. Chen, and S. K. Mitra, "Structure unanimity multiple description coding," *Signal Processing: Image Communication*, vol. 22, no. 1, pp. 59–68, Jan. 2007. doi: 10.1016/j.image.2006.11.003. [Online]. Available: http://dx.doi.org/10.1016/j.image.2006.11.003

[3] "MPEG DASH In a Nutshell." Bitcodin, 2015, Accessed: 2015-09. [Online]. Available: https://www.bitcodin.com/blog/2015/04/mpeg-dash/

[4] P. Reichl, B. Tuffin, and R. Schatz, "Logarithmic laws in service quality perception: where microeconomics meets psychophysics and quality of experience," *Telecommunication Systems*, pp. 587–600, Jun. 2011. doi: 10.1007/s11235-011-9503-7. [Online]. Available: http://dx.doi.org/10.1007/s11235-011-9503-7

[5] C. Chen, L. K. Choi, G. de Veciana, C. Caramanis, R. W. Heath, and A. C. Bovik, "A dynamic system model of time-varying subjective quality of video streams over HTTP," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, May 2013. doi: 10.1109/ICASSP.2013.6638329. ISBN 978-1-4799-0356-6 pp. 3602–3606. [Online]. Available: http://dx.doi.org/10.1109/ICASSP.2013.6638329

[6] D. P. Doane and L. E. Seward, "Measuring Skewness : A Forgotten Statistic?" *Journal of Statistics Education*, vol. 19, no. 2, pp. 1–18, 2011. doi: 10.1.1.362.5312. [Online]. Available: https://ww2.amstat.org/publications/jse/v19n2/doane.pdf

[7] A. J. S. Bernhard Schölkopf, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002. ISBN 9780262194754. [Online]. Available: https://books.google.pt/books?id=y8ORL3DWt4sC

[8] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol. 55, no. 10, p. 78, Oct. 2012. doi: 10.1145/2347736.2347755. [Online]. Available: http://dx.doi.org/10.1145/2347736.2347755

[9] D. Piech, "The State Of Online Video," pp. 1–41, 2010, Accessed: 09-2015. [Online]. Available: http://www.comscore.com/content/download/7235/125253/version/1/file/comScore+OMMA+Video+Presentation+-+Jan+2011.pdf

[10] S. J. Berman, B. Battino, and K. Feldman, "Beyond content: Capitalizing on the new revenue opportunities," 2010, Accessed: 09-2015. [Online]. Available: http://public.dhe.ibm.com/common/ssi/ecm/gb/en/gbe03361usen/GBE03361USEN.PDF

[11] Chao Chen, Lark Kwon Choi, G. de Veciana, C. Caramanis, R. W. Heath, and A. C. Bovik, "Modeling the Time—Varying Subjective Quality of HTTP Video Streams With Rate Adaptations," pp. 2206–2221, may 2014. [Online]. Available: http://dx.doi.org/10.1109/TIP.2014.2312613

[12] J. Greenwood, "The Third Industrial Revolution: Technology, Productivity, and Income Inequality," in *Economic Review*. The AEI Press, 1999, vol. 35, no. Q2, pp. 2–12. ISBN 0844770930. [Online]. Available: http://dx.doi.org/10.1109/S-Cube.2012.6225501

[13] Federal Communications Commission, "Open Internet Report and Order on Remand, Declaratory Ruling, and Order," *FCC 15-24*, p. 400, Feb. 2015. doi: 10.1017/CBO9781107415324.004. [Online]. Available: http://dx.doi.org/10.1017/CBO9781107415324.004

[14] Body of European Regulators for Electronic Communications, "BEREC Guidelines on the Implementation by National Regulators of European Net Neutrality Rules," p. 45, Aug. 2016. [Online]. Available: http://berec.europa.eu/eng/document_register/subject_matter/berec/download/0/6160-berec-guidelines-on-the-implementation-b_0.pdf

[15] Cisco, "Cisco Visual Networking Index : Forecast and Methodology, 2014 - 2019," Cisco Systems, Tech. Rep., 2015, Accessed: 2015-09. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.pdf

[16] P. (PWC), "Global entertainment and media outlook 2014-2018," 2014, Accessed: 2015-05. [Online]. Available: http://www.pwc.com/outlook

[17] B. Fung, "Verizon to Netflix: Here's a cease-and-desist letter. Can you hear me now?" *The Washington Post*, 2014, Accessed: 2015-01. [Online]. Available: http://www.washingtonpost.com/blogs/the-switch/wp/2014/06/05/verizon-to-netflix-heres-a-cease-and-desist-letter-can-you-hear-me-now/

[18] Marshall, Charlie, and F. Venturini, "Bringing TV to Life, Issue III TV is all around you," *Accenture*, no. III, p. 16, 2012, Accessed: 09-2015. [Online]. Available: https://www.accenture.com

[19] A. Metzger, K. Pohl, M. Papazoglou, E. di Nitto, A. Marconi, and D. Karastoyanova, "Research challenges on adaptive software and services in the future internet: towards an S-Cube research roadmap," in *Software Services and Systems Research - Results and Challenges (S-Cube), 2012 Workshop on European*, 2012. doi: 10.1109/S-Cube.2012.6225501. ISBN VO - pp. 1–7. [Online]. Available: http://dx.doi.org/10.1109/S-Cube.2012.6225501

[20] "NOTTS, Next-generation Over-The-Top Multimedia Services, Eureka! Celtic Plus C2012/2-4," 2015, Accessed: 2015-09. [Online]. Available: https://www.celticplus.eu/project-notts/

[21] "UltraTV - Ecossistema de aplicações para TV UltraHD, Portugal 2020 - 17738/2016," 2016.

[22] ISO/IEC, "Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats," in *ISO/IEC 23009-1:2012 - Information technology*, 2012, Accessed: 2015-09.

[23] G. Ibrahim, D. Kofman, Y. Chadli, and A. Ansiaux, "Toward a new Telco role in future content distribution services," in *2012 16th International Conference on Intelligence in Next Generation Networks*. IEEE, Oct. 2012. doi: 10.1109/ICIN.2012.6376029. ISBN 978-1-4673-1526-5 pp. 22–29. [Online]. Available: http://dx.doi.org/10.1109/ICIN.2012.6376029

[24] A. Kishore, "Operator CDNs : Making OTT Video Pay," *Verivue*, no. July, p. 14, 2011, Accessed: 09-2015. [Online]. Available: http://www.verivue.com/pdf/Operator_CDNs_Making_OTT_Pay.pdf

[25] J. Abreu, V. Becker, J. Nogueira, and B. Cardoso, "Time-shift services : a taxonomy and techno-business impacts of Catch-up TV," in *CENTERIS 2015 - Conference on ENTERprise Information Systems / PRojMAN 2015 - International Conference on Project MANagement / HCIST 2015 - International Conference on Health and Social Care Information Systems and Technologies*, 2015, p. 6.

[26] J. Nogueira, D. Gonzalez, L. Guardalben, and S. Sargento, "Over-The-Top Catch-up TV content-aware caching," in *2016 IEEE Symposium on Computers and Communication (ISCC)*. Messina, Italy: IEEE, jun 2016. doi: 10.1109/ISCC.2016.7543869. ISBN 978-1-5090-0679-3 pp. 1012–1017. [Online]. Available: http://dx.doi.org/10.1109/ISCC.2016.7543869

[27] J. Abreu, J. Nogueira, V. Becker, and B. Cardoso, "Survey of Catch-up TV and other time-shift services: a comprehensive analysis and taxonomy of linear and nonlinear television," *Telecommunication Systems*, Mar. 2016. doi: 10.1007/s11235-016-0157-3. [Online]. Available: http://dx.doi.org/10.1007/s11235-016-0157-3

[28] J. Nogueira, L. Guardalben, B. Cardoso, and S. Sargento, "Catch-up TV analytics: statistical characterization and consumption patterns identification on a production service," *Multimedia Systems*, vol. -, pp. 1–19, May 2016. doi: 10.1007/s00530-016-0516-7. [Online]. Available: http://dx.doi.org/10.1007/s00530-016-0516-7

[29] J. Nogueira, L. Guardalben, B. Cardoso, and S. Sargento, "Catch-up TV Forecasting: Enabling Next-Generation Over-The-Top Multimedia TV Services," *Multimedia Tools and Applications (Submitted)*, 2016.

[30] J. Nogueira, L. Guardalben, B. Cardoso, and S. Sargento, "Content-Aware Over-The-Top Delivery of Catch-up TV Services," *Transactions on Multimedia (To be submitted)*, 2016.

[31] A. Salvador, J. Nogueira, and S. Sargento, "QoE Assessment of HTTP Adaptive Video Streaming," 2015, pp. 235–242. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-18802-7_32

[32] "GAPOTT, Gravações Automáticas e Publicidade Over-The-Top, QREN SI I&DT 34009/2013," 2015, Accessed: 2015-09. [Online]. Available: http://gapott.ptinovacao.pt/

[33] D. Clark, "The design philosophy of the DARPA internet protocols," *ACM SIGCOMM Computer Communication Review*, vol. 18, no. 4, pp. 106–114, Aug. 1988. doi: 10.1145/52325.52336. [Online]. Available: http://dx.doi.org/10.1145/52325.52336

[34] J. H. Saltzer, D. P. Reed, and D. D. Clark, "End-to-end arguments in system design," *ACM Transactions on Computer Systems*, vol. 2, no. 4, pp. 277–288, Nov. 1984. doi: 10.1145/357401.357402. [Online]. Available: http://dx.doi.org/10.1145/357401.357402

[35] W. Cooper and G. Lovelace, "Delivering audio and video over broadband," in *IPTV Conference 2007 - Deployment and Service Delivery, IET*, 2007, pp. 1–66. ISBN 1905360126 Accessed: 09-2015. [Online]. Available: http://iptv-report.com/

[36] "Ericsson Mediaroom." Ericsson, 2015, Accessed: 2015-09. [Online]. Available: http://www.ericsson.com/us/ourportfolio/telecom-operators/mediaroom

[37] Microsoft Corporation, "Microsoft Smooth Streaming Protocol Specification," Microsoft Corporation, Tech. Rep., 2012, Accessed: 2015-09. [Online]. Available: http://www.iis.net/learn/media/smooth-streaming/smooth-streaming-transport-protocol

[38] Nielsen, "The Digital Consumer," pp. 1–28, 2014, Accessed: 09-2015. [Online]. Available: http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2014Reports/the-digital-consumer-report-feb-2014.pdf

[39] "Fast Forward: Implementing Live-to-VOD Services." Elemental, 2015, Accessed: 2015-09. [Online]. Available: http://www.digitaltveurope.net/309572/fast-forward-implementing-live-to-vod-services/

[40] "Content & Video delivery." Alcatel Lucent, 2015, Accessed: 2015-09. [Online]. Available: https://www.alcatel-lucent.com/products-solutions/content-delivery

[41] ANACOM, "Subscription Television Service Statistical Information 2nd Quarter 2015," ANACOM, Tech. Rep., 2015, Accessed: 12-2015. [Online]. Available: http://www.anacom.pt/render.jsp?contentId=1366001&languageId=1{#}.VoFTmBWLSHs

[42] CNC, "L ' économie de la télévision de rattrapage en 2014," *Centre national du cinéma et de l'image animée*, pp. 1–33, 2015. [Online]. Available: http://www.cnc.fr/web/fr/ressources/-/ressources/6592632

[43] H. Jenkins, *Convergence Culture: Where Old and New Media Collide*. NYU Press, 2006. ISBN 9780814743072. [Online]. Available: https://books.google.pt/books?id=RlRVNikT06YC

[44] Julian Clover, "Counting Netflix by country," 2014, Accessed: 09-2015. [Online]. Available: http://www.broadbandtvnews.com/2014/07/24/counting-netflix-by-country/

[45] Netflix, "Where is Netflix available," 2015, Accessed: 09-2015. [Online]. Available: https://help.netflix.com/en/node/14164

[46] F. Mann, R. Mahnke, and T. Hess, "Find Your Niches: A Guide for Managing Intermedia Effects Among Content Distribution Channels," *International Journal on Media Management*, vol. 14, no. 4, pp. 251–278, Oct. 2012. doi: 10.1080/14241277.2012.706763. [Online]. Available: http://dx.doi.org/10.1080/14241277.2012.706763

[47] E. Gruenwedel, "Comcast Controls 15% of EST Market, Other MVPDs to Enter," 2015, Accessed: 09-2015. [Online]. Available: http://www.homemediamagazine.com/digital-evolution/lionsgate-ceo-comcast-controls-15-est-market-other-mvpds-enter-32521

[48] Walt Disney Movies, "Video On Demand & Pay Per View," 2015, Accessed: 09-2015. [Online]. Available: http://disney.go.com/vodppv/vod.html

[49] R. Bury and J. Li, "Is it live or is it timeshifted, streamed or downloaded? Watching television in the era of multiple screens," *New Media & Society*, vol. 17, no. 4, pp. 592–610, Apr. 2015. doi: 10.1177/1461444813508368. [Online]. Available: http://dx.doi.org/10.1177/1461444813508368

[50] J. Vanattenhoven and D. Geerts, "Broadcast, Video-on-Demand, and Other Ways to Watch Television Content," in *Proceedings of the ACM International Conference on Interactive Experiences for TV and Online Video - TVX '15*. New York, New York, USA: ACM Press, 2015. doi: 10.1145/2745197.2745208. ISBN 9781450335263 pp. 73–82. [Online]. Available: http://dx.doi.org/10.1145/2745197.2745208

[51] T. Beauvisage and J.-S. Beuscart, "Audience dynamics of online catch up TV," in *Proceedings of the 21st international conference companion on World Wide Web - WWW '12 Companion*. New York, USA: ACM Press, 2012. doi: 10.1145/2187980.2188077. ISBN 9781450312301 p. 461. [Online]. Available: http://dx.doi.org/10.1145/2187980.2188077

[52] G. Nencioni, N. Sastry, J. Chandaria, J. Crowcroft, S. Nishanth, J. Chandaria, and J. Crowcroft, "Understanding and Decreasing the Network Footprint of Catch-up TV," in *Proceedings of the 22Nd International Conference on World Wide Web*. Rio de Janeiro, Brazil: International World Wide Web Conferences Steering Committee, 2013. ISBN 9781450320351 p. 12. [Online]. Available: http://dx.doi.org/10.1016/j.jnca.2012.08.014

[53] M. Cha, P. Rodriguez, J. Crowcroft, S. Moon, and X. Amatriain, "Watching television over an IP network," in *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement conference - IMC '08*, vol. 22. New York, New York, USA: ACM Press, 2008. doi: 10.1145/1452520.1452529. ISBN 9781605583341 p. 71. [Online]. Available: http://dx.doi.org/10.1145/1452520.1452529

[54] V. Gopalakrishnan, R. Jana, K. K. Ramakrishnan, D. F. Swayne, and V. A. Vaishampayan, "Understanding couch potatoes," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference - IMC '11*. New York, New York, USA: ACM Press, 2011. doi: 10.1145/2068816.2068838. ISBN 9781450310130 p. 225. [Online]. Available: http://dx.doi.org/10.1145/2068816.2068838

[55] F. Douglis and M. Kaashoek, "Scalable internet services," *IEEE Internet Computing*, vol. 5, no. 4, pp. 36–37, Jul. 2001. doi: 10.1109/MIC.2001.939448. [Online]. Available: http://dx.doi.org/10.1109/MIC.2001.939448

[56] A. Vakali and G. Pallis, "Content delivery networks: Status and trends," *Internet Computing, IEEE*, vol. 8, no. December, 2003. doi: 10.1109/MIC.2003.1250586. [Online]. Available: http://dx.doi.org/10.1109/MIC.2003.1250586

[57] G. Pallis and A. Vakali, "Insight and perspectives for content delivery networks," *Communications of the ACM*, vol. 49, no. 1, pp. 101–106, 2006. doi: 10.1145/1107458.1107462. [Online]. Available: http://dx.doi.org/10.1145/1107458.1107462

[58] S. Adler, "The Slashdot effect: an analysis of three Internet publications," *Linux Gazette*, no. 38, pp. 12–15, 1999, Accessed: 2015-09. [Online]. Available: http://linuxgazette.iatp.by/issue38/adler1.html

[59] T. Plagemann, V. Goebel, A. Mauthe, L. Mathy, T. Turletti, and G. Urvoy-Keller, "From content distribution networks to content networks — issues and challenges," *Computer Communications*, vol. 29, no. 5, pp. 551–562, Mar. 2006. doi: 10.1016/j.comcom.2005.06.006. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2005.06.006

[60] L. Wang, K. Park, and R. Pang, "Reliability and Security in the CoDeeN Content Distribution Network." in *Proceedings of the Annual Conference on USENIX Annual Technical Conference*. USENIX Association, 2004. [Online]. Available: http://dl.acm.org/citation.cfm?id=1247415.1247429

[61] M. Freedman, E. Freudenthal, and D. Mazieres, "Democratizing Content Publication with Coral." *NSDI*, 2004, Accessed: 2015-09. [Online]. Available: http://static.usenix.org/events/nsdi0/tech/full_papers/freedman/freedman_html/

[62] N. Yoshida, "Dynamic CDN Against Flash Crowds," in *Content Delivery Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 275–296. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-77887-5_11

[63] A. Pathan and R. Buyya, "A taxonomy and survey of content delivery networks," *Grid Computing and Distributed Systems (GRIDS) Laboratory*, pp. 1–44, 2007, Accessed: 2015-09. [Online]. Available: http://cloudbus.org/reports/CDN-Taxonomy.pdf

[64] "Akamai: Cloud Computing, Enterprise, Mobile, Security Solutions," Accessed: 2015-09. [Online]. Available: http://www.akamai.com

[65] M. Gordon and C. S. Officer, "A Powerful Trend that Represents Fundamental Change," Limelight Networks, Tech. Rep., 2006, Accessed: 2015-09. [Online]. Available: www.limelight.com

[66] K. W. Ross and V. Valloppillil, "Cache Array Routing Protocol (CARP) v1.0," INTERNET-DRAFT, Tech. Rep., 1998, Accessed: 2015-09. [Online]. Available: http://icp.ircache.net/carp.txt

[67] P. Vixie and D. Wessels, "Hyper Text Caching Protocol (HTCP/0.0)," Internet Engineering Task Force, Tech. Rep. January 2000, 2000, Accessed: 2015-09. [Online]. Available: http://www.ietf.org/rfc/rfc2756.txt

[68] "Cloudflare," Accessed: 2015-09. [Online]. Available: http://www.cloudflare.com

[69] L. Liu and F. Douglis, "Automatic fragment detection in dynamic Web pages and its impact on caching," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 859–874, Jun. 2005. doi: 10.1109/TKDE.2005.89. [Online]. Available: http://dx.doi.org/10.1109/TKDE.2005.89

[70] N. F. Mir, M. M. Nataraja, and S. Ravikrishnan, "A Performance Evaluation Study of Video-on-Demand Traffic over IP Networks," *2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications*, pp. 757–762, Mar. 2011. doi: 10.1109/WAINA.2011.102. [Online]. Available: http://dx.doi.org/10.1109/WAINA.2011.102

[71] V. Jacobson and R. Braden, "TCP Extensions for Long-Delay Paths," Internet Engineering Task Force, Tech. Rep., 1988. [Online]. Available: https://tools.ietf.org/html/rfc1072

[72] Community, "Popcorn Time," 2016, Accessed: 01-2016. [Online]. Available: http://popcorn-time.se/

[73] H. Ahlehagh and S. Dey, "Hierarchical video caching in wireless cloud: Approaches and algorithms," *2012 IEEE International Conference on Communications (ICC)*, pp. 7082–7087, Jun. 2012. doi: 10.1109/ICC.2012.6364966. [Online]. Available: http://dx.doi.org/10.1109/ICC.2012.6364966

[74] Netcraft, "November 2015 Web Server Survey," 2015, Accessed: 12-2015. [Online]. Available: http://news.netcraft.com/archives/2015/11/16/november-2015-web-server-survey.html

[75] The Apache Software Foundation, "Apache Traffic Server," 2015, Accessed: 12-2015. [Online]. Available: http://trafficserver.apache.org

[76] Microsoft Corporation, "Application Request Routing," 2015, Accessed: 12-2015. [Online]. Available: http://www.iis.net/downloads/microsoft/application-request-routing

[77] NGINX Inc., "NGINX High Performance Web Server," 2015, Accessed: 12-2015. [Online]. Available: http://nginx.com

[78] Varnish Software, "Varnish Cache," 2015, Accessed: 12-2015. [Online]. Available: https://www.varnish-software.com/

[79] D. Wessels, "Squid," 1996, Accessed: 01-2016. [Online]. Available: http://www.squid-cache.org/

[80] C. Bowman, P. B. Danzig, D. R. Hardy, U. Manber, and M. F. Schwartz, "The Harvest information discovery and access system," *Computer Networks and ISDN Systems*, vol. 28, no. 1-2, pp. 119–125, Dec. 1995. doi: 10.1016/0169-7552(95)00098-5. [Online]. Available: http://dx.doi.org/10.1016/0169-7552(95)00098-5

[81] P. B. Mirchandani and R. L. Francis, *Discrete Location Theory*. Wiley, Mar. 1990, vol. 24, no. 2. ISBN 978-0471892335. [Online]. Available: http://dx.doi.org/10.1002/net.3230240212

[82] Y. Bartal, "Probabilistic approximation of metric spaces and its algorithmic applications," *Proceedings of 37th Conference on Foundations of Computer Science*, pp. 184–193, 1996. doi: 10.1109/SFCS.1996.548477. [Online]. Available: http://dx.doi.org/10.1109/SFCS.1996.548477

[83] J. Hartigan, *Clustering Algorithms*, ser. Probability & Mathematical Statistics. John Wiley & Sons Inc, 1975. ISBN 978-0471356455. [Online]. Available: https://books.google.ca/books?id=v3fRAAAACAAJ

[84] Lili Qiu, V. Padmanabhan, and G. Voelker, "On the placement of Web server replicas," in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*, vol. 3. IEEE, 2001. doi: 10.1109/INFCOM.2001.916655. ISBN 0-7803-7016-3 pp. 1587–1596. [Online]. Available: http://dx.doi.org/10.1109/INFCOM.2001.916655

[85] P. Radoslavov, R. Govindan, and D. Estrin, "Topology-informed Internet replica placement," *Computer Communications*, vol. 25, no. 4, pp. 384–392, Mar. 2002. doi: 10.1016/S0140-3664(01)00410-8. [Online]. Available: http://dx.doi.org/10.1016/S0140-3664(01)00410-8

[86] P. Krishnan, D. Raz, and Y. Shavitt, "The cache location problem," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 568–582, 2000. doi: 10.1109/90.879344. [Online]. Available: http://dx.doi.org/10.1109/90.879344

[87] Y. Chen, R. H. Katz, and J. D. Kubiatowicz, "Dynamic Replica Placement for Scalable Content Delivery," in *Peer-to-Peer Systems*, 2002, pp. 306–318. [Online]. Available: http://dx.doi.org/10.1007/3-540-45748-8_29

[88] M. Karlsson and C. Karamanolis, "Choosing replica placement heuristics for wide-area systems," *24th International Conference on Distributed Computing Systems, 2004. Proceedings.*, pp. 350–359, 2004. doi: 10.1109/ICDCS.2004.1281600. [Online]. Available: http://dx.doi.org/10.1109/ICDCS.2004.1281600

[89] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl, "Globally distributed content delivery," *IEEE Internet Computing*, vol. 6, no. 5, pp. 50–58, Sep. 2002. doi: 10.1109/MIC.2002.1036038. [Online]. Available: http://dx.doi.org/10.1109/MIC.2002.1036038

[90] L.-c. Chen, "Approximation algorithms for data distribution with load balancing of web servers," *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pp. 274–281, 2001. doi: 10.1109/CLUSTR.2001.959988. [Online]. Available: http://dx.doi.org/10.1109/CLUSTR.2001.959988

[91] J. Kangasharju, J. Roberts, and K. W. Ross, "Object replication strategies in content distribution networks," *Computer Communications*, vol. 25, no. 4, pp. 376–383, Mar. 2002. doi: 10.1016/S0140-3664(01)00409-1. [Online]. Available: http://dx.doi.org/10.1109/MIC.2002.1036038

[92] R. Katz, "Efficient and adaptive web replication using content clustering," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 6, pp. 979–994, Aug. 2003. doi: 10.1109/JSAC.2003.814608. [Online]. Available: http://dx.doi.org/10.1109/JSAC.2003.814608

[93] N. Fujita, Y. Ishikawa, A. Iwata, and R. Izmailov, "Coarse-grain replica management strategies for dynamic replication of Web contents," *Computer Networks*, vol. 45, no. 1, pp. 19–34, May 2004. doi: 10.1016/j.comnet.2004.02.006. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2004.02.006

[94] K. Johnson, J. Carr, M. Day, and M. Kaashoek, "The measured performance of content distribution networks," *Computer Communications*, vol. 24, no. 2, pp. 202–206, Feb. 2001. doi: 10.1016/S0140-3664(00)00315-7. [Online]. Available: http://dx.doi.org/10.1016/S0140-3664(00)00315-7

[95] M. Freedman and D. Mazieres, "Sloppy hashing and self-organizing clusters," *Peer-to-Peer Systems II*, pp. 1–6, 2003. doi: 10.1007/978-3-540-45172-3_4. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-45172-3_4

[96] A. Leff, J. Wolf, and P. Yu, "Replication algorithms in a remote caching architecture," *IEEE Transactions on Parallel and Distributed Systems*, vol. 4, no. 11, pp. 1185–1204, 1993. doi: 10.1109/71.250099. [Online]. Available: http://dx.doi.org/10.1109/71.250099

[97] I. Cidon, S. Kutten, and R. Soffer, "Optimal allocation of electronic content," *Computer Networks*, vol. 40, no. 2, pp. 205–218, Oct. 2002. doi: 10.1016/S1389-1286(02)00251-7. [Online]. Available: http://dx.doi.org/10.1016/S1389-1286(02)00251-7

[98] Bong-Jun Ko and D. Rubenstein, "Distributed, self-stabilizing placement of replicated resources in emerging networks," in *11th IEEE International Conference on Network Protocols, 2003. Proceedings.* IEEE Comput. Soc, 2003. doi: 10.1109/ICNP.2003.1249752. ISBN 0-7695-2024-3 pp. 6–15. [Online]. Available: http://dx.doi.org/10.1109/ICNP.2003.1249752

[99] S. Tse, "Approximate algorithms for document placement in distributed Web servers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 6, pp. 489–496, Jun. 2005. doi: 10.1109/TPDS.2005.63. [Online]. Available: http://dx.doi.org/10.1109/TPDS.2005.63

[100] B. Wu and A. Kshemkalyani, "Objective-optimal algorithms for long-term Web prefetching," *IEEE Transactions on Computers*, vol. 55, no. 1, pp. 2–17, Jan. 2006. doi: 10.1109/TC.2006.12. [Online]. Available: http://dx.doi.org/10.1109/TC.2006.12

[101] G. Pallis, a. Vakali, K. Stamos, a. Sidiropoulos, D. Katsaros, and Y. Manolopoulos, "A Latency-Based Object Placement Approach in Content Distribution Networks," *Third Latin American Web Congress (LA-WEB'2005)*, pp. 140–147, 2005. doi: 10.1109/LAWEB.2005.3. [Online]. Available: http://dx.doi.org/10.1109/LAWEB.2005.3

[102] N. Laoutaris, V. Zissimopoulos, and I. Stavrakakis, "On the optimization of storage capacity allocation for content distribution," *Computer Networks*, vol. 32686, pp. 1–22, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128604002403

[103] T. Bektaş, J.-F. Cordeau, E. Erkut, and G. Laporte, "Exact algorithms for the joint object placement and request routing problem in content distribution networks," *Computers & Operations Research*, vol. 35, no. 12, pp. 3860–3884, Dec. 2008. doi: 10.1016/j.cor.2007.02.005. [Online]. Available: http://dx.doi.org/10.1016/j.cor.2007.02.005

[104] J. Challenger, P. Dantzig, A. Iyengar, and K. Witting, "A fragment-based approach for efficiently creating dynamic web content," *ACM Transactions on Internet Technology*, vol. 5, no. 2, pp. 359–389, May 2005. doi: 10.1145/1064340.1064343. [Online]. Available: http://dx.doi.org/10.1145/1064340.1064343

[105] S. Bakiras and T. Loukopoulos, "Combining replica placement and caching techniques in content distribution networks," *Computer Communications*, vol. 28, no. 9, pp. 1062–1073, Jun. 2005. doi: 10.1016/j.comcom.2005.01.012. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2005.01.012

[106] K. Stamos, G. Pallis, and A. Vakali, "Integrating caching techniques on a content distribution network," *Advances in Databases and Information Systems*, pp. 200–215, 2006. doi: 10.1007/11827252_17. [Online]. Available: http://dx.doi.org/10.1007/11827252_17

[107] M. Abrams, C. R. Standridge, G. Abdulla, E. A. Fox, and S. Williams, "Removal policies in network caches for World-Wide Web documents," *ACM SIGCOMM Computer Communication Review*, vol. 26, no. 4, pp. 293–305, Oct. 1996. doi: 10.1145/248157.248182. [Online]. Available: http://dx.doi.org/10.1145/248157.248182

[108] C. Canali, V. Cardellini, M. Colajanni, and R. Lancellotti, "Content Delivery and Management," in *Content Delivery Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 105–126. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-77887-5_4

[109] K. Stamos, G. Pallis, and A. Vakali, "Caching Techniques on CDN Simulated Frameworks," in *Content Delivery Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 127–153. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-77887-5_5

[110] D. Wessels and K. Claffy, "IETF RFC 2186: Internet Cache Protocol (ICP), version 2," Sep. 1997, Accessed: 2015-09. [Online]. Available: https://www.ietf.org/rfc/rfc2186.txt

[111] A. Rousskov and D. Wessels, "Cache Digests," *Computer Networks and ISDN Systems*, pp. 1–19, 1998. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.4810

[112] S. Gadde, M. Rabinovich, and J. Chase, "Reduce , Reuse , Recycle : An Approach to Building Large Internet Caches," *The Sixth Workshop on Hot Topics in Operating Systems*, 1997.

[113] D. Karger, A. Sherman, A. Berkheimer, B. Bogstad, R. Dhanidina, K. Iwamoto, B. Kim, L. Matkins, and Y. Yerushalmi, "Web caching with consistent hashing," *Computer Networks*, vol. 31, no. 11-16, pp. 1203–1213, May 1999. doi: 10.1016/S1389-1286(99)00055-9. [Online]. Available: http://dx.doi.org/10.1016/S1389-1286(99)00055-9

[114] D. Tsang, "Large-scale cooperative caching and application-level multicast in multimedia content delivery networks," *IEEE Communications Magazine*, vol. 43, no. 5, pp. 98–105, May 2005. doi: 10.1109/MCOM.2005.1453429. [Online]. Available: http://dx.doi.org/10.1109/MCOM.2005.1453429

[115] Jian Ni, D. Tsang, I. Yeung, and Xiaojun Hei, "Hierarchical content routing in large-scale multimedia content delivery network," in *IEEE International Conference on Communications, 2003. ICC '03.*, vol. 2. IEEE, 2003. doi: 10.1109/ICC.2003.1204454. ISBN 0-7803-7802-4 pp. 854–859. [Online]. Available: http://dx.doi.org/10.1109/ICC.2003.1204454

[116] Y. Suwa and O. Altintas, "Scalable Request Routing with Next-Neighbor Load Sharing in Multi-Server Environments," *19th International Conference on Advanced Information Networking and Applications (AINA'05) Volume 1 (AINA papers)*, vol. 1, pp. 441–446, 2005. doi: 10.1109/AINA.2005.303. [Online]. Available: http://dx.doi.org/10.1109/AINA.2005.303

[117] S. Sivasubramanian, M. Szymaniak, G. Pierre, and M. V. Steen, "Replication for web hosting systems," *ACM Computing Surveys*, vol. 36, no. 3, pp. 291–334, Sep. 2004. doi: 10.1145/1035570.1035573. [Online]. Available: http://dx.doi.org/10.1145/1035570.1035573

[118] Broadpeak, "umbrellaCDN," Broadpeak, Rennes, France, Tech. Rep., 2013, Accessed: 2015-09. [Online]. Available: http://www.broadpeak.tv/upload/produit/fichier/29-798-broadpeak_umbrellacdn_datasheet.pdf

[119] L. Wang, V. Pai, and L. Peterson, "The effectiveness of request redirection on CDN robustness," *ACM SIGOPS Operating Systems Review*, pp. 345–360, 2002. [Online]. Available: http://dl.acm.org/citation.cfm?id=844160

[120] V. S. Pai, M. Aron, G. Banga, M. Svendsen, P. Druschel, W. Zwaenepoel, and E. Nahum, "Locality-aware request distribution in cluster-based network servers," *ACM SIGPLAN Notices*, vol. 33, no. 11, pp. 205–216, Nov. 1998. doi: 10.1145/291006.291048. [Online]. Available: http://dx.doi.org/10.1145/291006.291048

[121] M. Szymaniak, G. Pierre, and M. van Steen, "Netairt: A Flexible Redirection System for Apache." *International Conference WWW/Internet (ICWI)*, 2003. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.78.927&rep=rep1&type=pdf

[122] G. Pierre and M. van Steen, "Globule: a collaborative content delivery network," *IEEE Communications Magazine*, vol. 44, no. 8, pp. 127–133, Aug. 2006. doi: 10.1109/MCOM.2006.1678120. [Online]. Available: http://dx.doi.org/10.1109/MCOM.2006.1678120

[123] N. Ball and P. Pietzuch, "Distributed content delivery using load-aware network coordinates," *Proceedings of the 2008 ACM CoNEXT Conference on - CONEXT '08*, pp. 1–6, 2008. doi: 10.1145/1544012.1544089. [Online]. Available: http://dx.doi.org/10.1145/1544012.1544089

[124] D. Starobinski, "A Comparative Analysis of Server Selection in Content Replication Networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 6, pp. 1461–1474, Dec. 2008. doi: 10.1109/TNET.2007.909752. [Online]. Available: http://dx.doi.org/10.1109/TNET.2007.909752

[125] B. Huffaker, M. Fomenkov, D. J. Plummer, and D. Moore, "Distance Metrics in the Internet," *IEEE International Telecommunications Symposium*, pp. 1–6, 2002. [Online]. Available: http://www.caida.org/outreach/presentations/2002/Distance/its2002.pdf

[126] M. Conti, E. Gregori, and W. Lapenna, "Content Delivery Policies in Replicated Web Services: Client-Side vs. Server-Side," *Cluster Computing*, vol. 8, no. 1, pp. 47–60, Jan. 2005. doi: 10.1007/s10586-004-4436-5. [Online]. Available: http://dx.doi.org/10.1007/s10586-004-4436-5

[127] M. Conti, E. Gregori, and W. Lapenna, "Client-side content delivery policies in replicated web services: parallel access versus single server approach," *Performance Evaluation*, vol. 59, no. 2-3, pp. 137–157, Feb. 2005. doi: 10.1016/j.peva.2004.07.018. [Online]. Available: http://dx.doi.org/10.1016/j.peva.2004.07.018

[128] J. Pan, Y. Hou, and B. Li, "An overview of DNS-based server selections in content distribution networks," *Computer Networks*, vol. 43, no. 6, pp. 695–711, Dec. 2003. doi: 10.1016/S1389-1286(03)00293-7. [Online]. Available: http://dx.doi.org/10.1016/S1389-1286(03)00293-7

[129] A. Su, D. R. Choffnes, A. Kuzmanovic, and F. E. Bustamante, "Drafting behind Akamai (travelocity-based detouring)," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, p. 435, Aug. 2006. doi: 10.1145/1151659.1159962. [Online]. Available: http://dx.doi.org/10.1145/1151659.1159962

[130] B. Cain, R. Nair, and O. Spatscheck, "Known Content Network (CN) Request-Routing Mechanisms," Internet Engineering Task Force RFC 3568, Tech. Rep., 2003, Accessed: 2015-09. [Online]. Available: http://tools.ietf.org/html/rfc3568

[131] M. Hofmann and L. R. Beaumont, *Content Networking: Architecture, Protocols, and Practice*, 1st ed. Elsevier, 2005. ISBN 978-1558608344

[132] M. Freedman, K. Lakshminarayanan, and D. Mazières, "OASIS: Anycast for Any Service." *NSDI'06 Proceedings of the 3rd conference on Networked Systems Design & Implementation*, vol. 3, 2006. [Online]. Available: https://www.usenix.org/legacy/events/nsdi06/tech/full_papers/freedman/freedman_html/

[133] R. Buyya, A.-m. Pathan, J. Broberg, and Z. Tari, "A Case for Peering of Content Delivery Networks," *IEEE Distributed Systems Online*, vol. 7, no. 10, pp. 3–3, Oct. 2006. doi: 10.1109/MDSO.2006.57. [Online]. Available: http://dx.doi.org/10.1109/MDSO.2006.57

[134] R. Krishnan, H. V. Madhyastha, S. Srinivasan, S. Jain, A. Krishnamurthy, T. Anderson, and J. Gao, "Moving beyond end-to-end path information to optimize CDN performance," *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference - IMC '09*, p. 190, 2009. doi: 10.1145/1644893.1644917. [Online]. Available: http://dx.doi.org/10.1145/1644893.1644917

[135] P. Sun, M. Yu, M. J. Freedman, and J. Rexford, "Identifying performance bottlenecks in CDNs through TCP-level monitoring," in *Proceedings of the first ACM SIGCOMM workshop on Measurements up the stack - W-MUST '11*. New York, New York, USA: ACM Press, 2011. doi: 10.1145/2018602.2018615. ISBN 9781450308007 p. 49. [Online]. Available: http://dx.doi.org/10.1145/2018602.2018615

[136] "Keynote Systems - Web and Mobile Service Performance Testing Corporation,," Accessed: 2015-09. [Online]. Available: http://www.keynote.com/

[137] "WebPagetest," Accessed: 2015-09. [Online]. Available: http://www.webpagetest.org/

[138] K. Stamos, G. Pallis, A. Vakali, D. Katsaros, A. Sidiropoulos, and Y. Manolopoulos, "CDNsim," *ACM Transactions on Modeling and Computer Simulation*, vol. 20, no. 2, pp. 1–40, Apr. 2010. doi: 10.1145/1734222.1734226. [Online]. Available: http://dx.doi.org/10.1145/1734222.1734226

[139] H. Abrahamsson and M. Bjorkman, "Caching for IPTV distribution with time-shift," in *2013 International Conference on Computing, Networking and Communications (ICNC)*. San Diego, CA: IEEE, Jan. 2013. doi: 10.1109/ICCNC.2013.6504212. ISBN 978-1-4673-5288-8 pp. 916–921. [Online]. Available: http://dx.doi.org/10.1109/ICCNC.2013.6504212

[140] Visible Measures Research, "Viewer Abandonment Trends in Short-Form Online Video Content," Visible Measures Research, Tech. Rep. 7, 2012. [Online]. Available: http://corp.visiblemeasures.com/Portals/382/docs/videoabandonmentresearch.pdf

[141] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "Youtube traffic characterization," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement - IMC '07*. New York, New York, USA: ACM Press, 2007. doi: 10.1145/1298306.1298310. ISBN 9781595939081 p. 15. [Online]. Available: http://dx.doi.org/10.1145/1298306.1298310

[142] T. Lohmar, T. Einarsson, P. Frojdh, F. Gabin, and M. Kampmann, "Dynamic adaptive HTTP streaming of live content," in *2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*. IEEE, Jun. 2011. doi: 10.1109/WoWMoM.2011.5986186. ISBN 978-1-4577-0352-2 pp. 1–8. [Online]. Available: http://dx.doi.org/10.1109/WoWMoM.2011.5986186

[143] R. Pantos, "HTTP Live Streaming," in *IETF Internet-Draft*, 2012, Accessed: 2015-09. [Online]. Available: http://tools.ietf.org/html/draft-pantos-http-live-streaming-10

[144] M. Czyrnek, E. Kusmierek, C. Mazurek, M. Stroinski, and J. Weglarz, "CDN for Live and On-Demand Video Services over IP," in *Content Delivery Networks*, ser. Lecture Notes Electrical Engineering, R. Buyya, M. Pathan, and A. Vakali, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, vol. 9, pp. 317–342. ISBN 978-3-540-77886-8. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-77887-5

[145] M. Czyrnek, E. Kusmierek, C. Mazurek, and M. Stroinski, "New Services for iTVP Content Providers to Manage Live and On-Demand Content Streaming," *2008 International Conference on Automated Solutions for Cross Media Content and Multi-Channel Distribution*, pp. 180–186, Nov. 2008. doi: 10.1109/AXMEDIS.2008.28. [Online]. Available: http://dx.doi.org/10.1109/AXMEDIS.2008.28

[146] R. Ranjan, B. Benatallah, S. Dustdar, and M. P. Papazoglou, "Cloud Resource Orchestration Programming: Overview, Issues, and Directions," *IEEE Internet Computing*, vol. 19, no. 5, pp. 46–56, Sep. 2015. doi: 10.1109/MIC.2015.20. [Online]. Available: http://dx.doi.org/10.1109/MIC.2015.20

[147] Y. Kryftis, G. Mastorakis, C. X. Mavromoustakis, J. M. Batalla, E. Pallis, and G. Kormentzas, "Efficient entertainment services provision over a novel network architecture," *IEEE Wireless Communications*, vol. 23, no. 1, pp. 14–21, Feb. 2016. doi: 10.1109/MWC.2016.7422401. [Online]. Available: http://dx.doi.org/10.1109/MWC.2016.7422401

[148] R. Weingärtner, G. B. Bräscher, and C. B. Westphall, "Cloud resource management: A survey on forecasting and profiling models," *Journal of Network and Computer Applications*, vol. 47, pp. 99–106, Jan. 2015. doi: 10.1016/j.jnca.2014.09.018. [Online]. Available: http://dx.doi.org/10.1016/j.jnca.2014.09.018

[149] J. Kephart and D. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, Jan. 2003. doi: 10.1109/MC.2003.1160055. [Online]. Available: http://dx.doi.org/10.1109/MC.2003.1160055

[150] F. Bahrpeyma, H. Haghighi, and A. Zakerolhosseini, "An adaptive RL based approach for dynamic resource provisioning in Cloud virtualized data centers," *Computing*, vol. 97, no. 12, pp. 1209–1234, Dec. 2015. doi: 10.1007/s00607-015-0455-8. [Online]. Available: http://dx.doi.org/10.1007/s00607-015-0455-8

[151] H. Koumaras, D. Negru, E. Borcoci, V. Koumaras, C. Troulos, Y. Lapid, E. Pallis, M. Sidibé, A. Pinto, G. Gardikis, G. Xilouris, and C. Timmerer, "Media Ecosystems: A Novel Approach for Content-Awareness in Future Networks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011, vol. 6656, pp. 369–380. ISBN 9783642208973. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-20898-0_26

[152] H. de Meer, K. A. Hummel, and R. Basmadjian, "Future Internet services and architectures: trends and visions," *Telecommunication Systems*, vol. 51, no. 4, pp. 219–220, Dec. 2012. doi: 10.1007/s11235-011-9430-7. [Online]. Available: http://dx.doi.org/10.1007/s11235-011-9430-7

[153] P. Świątek, K. Juszczyszyn, K. Brzostowski, J. Drapała, and A. Grzech, "Supporting Content, Context and User Awareness in Future Internet Applications," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012, vol. 7281 LNCS, pp. 154–165. ISBN 9783642302404. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-30241-1_14

[154] M. M. Amble, P. Parag, S. Shakkottai, and L. Ying, "Content-aware caching and traffic management in content distribution networks," in *2011 Proceedings IEEE INFOCOM*, no. D. IEEE, Apr. 2011. doi: 10.1109/INFCOM.2011.5935123. ISBN 978-1-4244-9919-9. ISSN 0743166X pp. 2858–2866. [Online]. Available: http://dx.doi.org/10.1109/INFCOM.2011.5935123

[155] J. Mongay Batalla, A. Beben, and Y. Chen, "Optimized decision algorithm for Information Centric Networks," *Telecommunication Systems*, vol. 61, no. 2, pp. 247–255, 2016. doi: 10.1007/s11235-015-9998-4. [Online]. Available: http://dx.doi.org/10.1007/s11235-015-9998-4

[156] M. Mangili, F. Martignon, A. Capone, and F. Malucelli, "Content-aware planning models for information-centric networking," in *2014 IEEE Global Communications Conference*. IEEE, Dec. 2014. doi: 10.1109/GLOCOM.2014.7037078. ISBN 978-1-4799-3512-3 pp. 1854–1860. [Online]. Available: http://dx.doi.org/10.1109/GLOCOM.2014.7037078

[157] A. Tiwari and P. Kanungo, "Dynamic load balancing algorithm for scalable heterogeneous web server cluster with content awareness," in *Trendz in Information Sciences & Computing(TISC2010)*. IEEE, Dec. 2010. doi: 10.1109/TISC.2010.5714626. ISBN 978-1-4244-9007-3 pp. 143–148. [Online]. Available: http://dx.doi.org/10.1109/TISC.2010.5714626

[158] H. Mihara, D. Imachi, M. Yamamoto, T. Miyamura, and K. Sasayama, "Content aware routing: A content oriented traffic engineering," in *2013 IEEE Global Communications Conference (GLOBECOM)*. IEEE, Dec. 2013. doi: 10.1109/GLOCOM.2013.6831272. ISBN 978-1-4799-1353-4 pp. 1416–1421. [Online]. Available: http://dx.doi.org/10.1109/GLOCOM.2013.6831272

[159] A. BR, L. Reddy, P. Hiremath, and N. SS, "RTSP Audio and Video Streaming for QoS in Wireless Mobile Devices," *IJCSNS*, vol. 8, no. 1, pp. 96–101, 2008. [Online]. Available: http://paper.ijcsns.org/07_book/200801/20080114.pdf

[160] Multiparty Multimedia Session Control Working Group of the Internet Engineering Task Force, "Real Time Streaming Protocol," in *RFC 2326*, 1998, Accessed: 2015-09. [Online]. Available: http://tools.ietf.org/html/rfc2326

[161] A. Begen, T. Akgul, and M. Baugher, "Watching Video over the Web: Part 1: Streaming Protocols," *IEEE Internet Computing*, vol. 15, no. 2, pp. 54–63, Mar. 2011. doi: 10.1109/MIC.2010.155. [Online]. Available: http://dx.doi.org/10.1109/MIC.2010.155

[162] Audio-Video Transport Working Group of the Internet Engineering Task Force, "RTP: A Transport Protocol for Real-Time Applications," in *RFC 1889*, 1996, Accessed: 2015-09. [Online]. Available: http://tools.ietf.org/html/rfc1889

[163] Audio-Video Transport Working Group of the Internet Engineering Task Force, "RTP: A Transport Protocol for Real-Time Applications," in *RFC 3550*, 2003, Accessed: 2015-09. [Online]. Available: http://tools.ietf.org/html/rfc3550

[164] J. Rosenberg and H. Schulzrinne, "Timer reconsideration for enhanced RTP scalability," *Proceedings. IEEE INFOCOM '98, the Conference on Computer Communications. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Gateway to the 21st Century (Cat. No.98CH36169)*, vol. 1, pp. 233–241, 1998. doi: 10.1109/INFCOM.1998.659659. [Online]. Available: http://dx.doi.org/10.1109/INFCOM.1998.659659

[165] K. J. Ma, M. Mikhailov, and R. Bartos, "DRM optimization for stitched media file rate adaptation," in *2011 IEEE International Conference on Multimedia and Expo*, vol. 1. IEEE, Jul. 2011. doi: 10.1109/ICME.2011.6012012. ISBN 978-1-61284-348-3 pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/ICME.2011.6012012

[166] S. Sivasothy, G. Lee, and N. Crespi, "A unified session control protocol for IPTV services," *11th Advanced Communication Technology, 2009. ICACT.*, pp. 961–965, 2009. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4809574

[167] T. Wauters, W. Van de Meerssche, F. De Turck, B. Dhoedt, P. Demeester, T. Van Caenegem, and E. Six, "Co-operative Proxy Caching Algorithms for Time-Shifted IPTV Services," *32nd EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO'06)*, pp. 379–386, 2006. doi: 10.1109/EUROMICRO.2006.29. [Online]. Available: http://dx.doi.org/10.1109/EUROMICRO.2006.29

[168] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang, "Understanding the impact of video quality on user engagement," in *Proceedings of the ACM SIGCOMM 2011 conference on SIGCOMM - SIGCOMM '11*. New York, USA: ACM Press, 2011. doi: 10.1145/2018436.2018478. ISBN 9781450307970 pp. 362–373. [Online]. Available: http://dx.doi.org/10.1145/2018436.2018478

[169] Y. Sanchez, T. Schierl, C. Hellge, T. Wiegand, D. Hong, D. De Vleeschauwer, W. Van Leekwijck, and Y. Le Louédec, "Efficient HTTP-based streaming using Scalable Video Coding," *Signal Processing: Image Communication*, vol. 27, no. 4, pp. 329–342, Apr. 2012. doi: 10.1016/j.image.2011.10.002. [Online]. Available: http://dx.doi.org/10.1016/j.image.2011.10.002

[170] V. Goyal, "Multiple description coding: compression meets the network," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 74–93, 2001. doi: 10.1109/79.952806. [Online]. Available: http://dx.doi.org/10.1109/79.952806

[171] A. L. Vitali, A. Borneo, M. Fumagalli, and R. Rinaldo, "Video over IP using standard-compatible multiple description coding: An IETF proposal," *Journal of Zhejiang University SCIENCE A*, vol. 7, no. 5, pp. 668–676, May 2006. doi: 10.1631/jzus.2006.A0668. [Online]. Available: http://dx.doi.org/10.1631/jzus.2006.A0668

[172] A. Zhao, W. Wang, H. Cui, and K. Tang, "Efficient multiple description scalable video coding scheme based on weighted signal combinations," *Tsinghua Science and Technology*, vol. 12, no. 1, pp. 86–90, Feb. 2007. doi: 10.1016/S1007-0214(07)70013-5. [Online]. Available: http://dx.doi.org/10.1016/S1007-0214(07)70013-5

[173] N. Franchi, M. Fumagalli, R. Lancini, and S. Tubaro, "Multiple description video coding for scalable and robust transmission over IP," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 3, pp. 321–334, Mar. 2005. doi: 10.1109/TCSVT.2004.842606. [Online]. Available: http://dx.doi.org/10.1109/TCSVT.2004.842606

[174] ITU-T and ISO/IEC, "ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC)," in *Advanced Video Coding for Generic Audiovisual Services*, 2007, Accessed: 2015-09.

[175] T. Schierl, T. Stockhammer, and T. Wiegand, "Mobile Video Transmission Using Scalable Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1204–1217, Sep. 2007. doi: 10.1109/TCSVT.2007.905528. [Online]. Available: http://dx.doi.org/10.1109/TCSVT.2007.905528

[176] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H. 264/AVC standard," *IEEE Transactions on Circuits and Systems for Video*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007. doi: 10.1109/TCSVT.2007.905532. [Online]. Available: http://dx.doi.org/10.1109/TCSVT.2007.905532

[177] A. Fox, S. Gribble, Y. Chawathe, and E. Brewer, "Adapting to network and client variation using infrastructural proxies: lessons and perspectives," *IEEE Personal Communications*, vol. 5, no. 4, pp. 10–19, 1998. doi: 10.1109/98.709365. [Online]. Available: http://dx.doi.org/10.1109/98.709365

[178] B. Badrinath, A. Fox, L. Kleinrock, G. Popek, P. Reiher, and M. Satyanarayanan, "A conceptual framework for network and client adaptation," *Mobile Networks and Applications*, vol. 5, pp. 221–231, 2000. doi: 10.1023/A:1019168830964. [Online]. Available: http://dx.doi.org/10.1023/A:1019168830964

[179] Will Law, "DASHing Into an Era of Convergence," 2012, Accessed: 2015-09. [Online]. Available: https://blogs.akamai.com/2012/04/san-francisco-has-a-largely.html

[180] ISO/IEC, "Coding of audio-visual objects – Part 12: ISO base media file format," in *ISO/IEC 14496-12:2008 - Information technology*, 2008, Accessed: 2015-09.

[181] Microsoft, "Portable encoding of audio-video objects," in *The Protected Interoperable File Format (PIFF)*, 2009, Accessed: 2015-09. [Online]. Available: http://go.microsoft.com/?linkid=9682897

[182] "VLC: Open-Source Multimedia Framework, Player and Server," Accessed: 2015-09. [Online]. Available: http://www.videolan.org/vlc/

[183] "Adobe Flash Video File Format Specification Version 10.1," Adobe Systems Incorporated, Tech. Rep., 2010, Accessed: 2015-09. [Online]. Available: http://download.macromedia.com/f4v/video_file_format_spec_v10_1.pdf

[184] "Flash Media Manifest File Format Specification," Adobe Systems Incorporated, Tech. Rep., 2010, Accessed: 2015-09. [Online]. Available: http://osmf.org/dev/osmf/specpdfs/FlashMediaManifestFileFormatSpecification.pdf

[185] "Wowza: Media Server & Video Streaming Server," Accessed: 2015-09. [Online]. Available: http://www.wowza.com

[186] 3GPP - 3rd Generation Partnership Project, "Release 10," 2011, Accessed: 2015-09. [Online]. Available: http://www.3gpp.org/Release-10

[187] 3GPP - 3rd Generation Partnership Project, "Release 11," 2012, Accessed: 2015-09. [Online]. Available: http://www.3gpp.org/Release-11

[188] ISO/IEC, "MPEG systems technologies – Part 7: Common encryption in ISO base media file format files," in *ISO/IEC 23001-7:2012 - Information technology*, 2012, Accessed: 2015-09.

[189] P. Ni, A. Eichhorn, C. Griwodz, and P. l. Halvorsen, "Fine-grained scalable streaming from coarse-grained videos," in *Proceedings of the 18th international workshop on Network and operating systems support for digital audio and video - NOSSDAV '09*. New York, New York, USA: ACM Press, 2009. doi: 10.1145/1542245.1542269. ISBN 9781605584331 p. 103. [Online]. Available: http://dx.doi.org/10.1145/1542245.1542269

[190] M. Zink, O. Künzel, J. Schmitt, and R. Steinmetz, "Subjective impression of variations in layer encoded videos," *IWQoS'03 Proceedings of the 11th international conference on Quality of service*, pp. 137–154, 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=1784048

[191] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP," *Proceedings of the second annual ACM conference on Multimedia systems - MMSys '11*, p. 157, 2011. doi: 10.1145/1943552.1943574. [Online]. Available: http://dx.doi.org/10.1145/1943552.1943574

[192] H. Riiser, T. Endestad, P. Vigmostad, C. Griwodz, and P. Halvorsen, "Video streaming using a location-based bandwidth-lookup service for bitrate planning," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 8, no. 3, pp. 1–19, Jul. 2012. doi: 10.1145/2240136.2240137. [Online]. Available: http://dx.doi.org/10.1145/2240136.2240137

[193] H. Riiser, P. Halvorsen, C. Griwodz, and B. Hestnes, "Performance measurements and evaluation of video streaming in HSDPA networks with 16QAM modulation," in *2008 IEEE International Conference on Multimedia and Expo*, vol. 1. IEEE, Jun. 2008. doi: 10.1109/ICME.2008.4607478. ISBN 978-1-4244-2570-9 pp. 489–492. [Online]. Available: http://dx.doi.org/10.1109/ICME.2008.4607478

[194] H. Riiser, H. k. S. Bergsaker, P. Vigmostad, P. l. Halvorsen, and C. Griwodz, "A comparison of quality scheduling in commercial adaptive HTTP streaming solutions on a 3G network," *Proceedings of the 4th Workshop on Mobile Video - MoVid '12*, p. 25, 2012. doi: 10.1145/2151677.2151684. [Online]. Available: http://dx.doi.org/10.1145/2151677.2151684

[195] A. Goel, C. Krasic, and J. Walpole, "Low-latency adaptive streaming over tcp," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 4, no. 3, pp. 1–20, Aug. 2008. doi: 10.1145/1386109.1386113. [Online]. Available: http://dx.doi.org/10.1145/1386109.1386113

[196] L. De Cicco, S. Mascolo, and V. Palmisano, "Feedback control for adaptive live video streaming," in *Proceedings of the second annual ACM conference on Multimedia systems - MMSys '11*. New York, New York, USA: ACM Press, 2011. doi: 10.1145/1943552.1943573. ISBN 9781450305181 p. 145. [Online]. Available: http://dx.doi.org/10.1145/1943552.1943573

[197] A. Tirumala, M. Gates, F. Qin, J. Dugan, and J. Ferguson, "Iperf - The TCP/UDP bandwidth measurement tool," Accessed: 2015-09. [Online]. Available: http://iperf.sourceforge.net/

[198] A. Balamash and M. Krunz, "An overview of web caching replacement algorithms," *IEEE Communications Surveys & Tutorials*, vol. 6, no. 2, pp. 44–56, 2004. doi: 10.1109/COMST.2004.5342239. [Online]. Available: http://dx.doi.org/10.1109/COMST.2004.5342239

[199] E. J. O'Neil, P. E. O'Neil, and G. Weikum, "The LRU-K page replacement algorithm for database disk buffering," *ACM SIGMOD Record*, vol. 22, no. 2, pp. 297–306, Jun. 1993. doi: 10.1145/170036.170081. [Online]. Available: http://dx.doi.org/10.1145/170036.170081

[200] J.-M. Menaud, V. Issarny, and M. Banâtre, "Improving the Effectiveness of Web Caching," in *Advances in Distributed Systems: Advanced Distributed Computing: From Algorithms to Systems*, S. Krakowiak and S. Shrivastava, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 375–401. ISBN 978-3-540-46475-4. [Online]. Available: http://dx.doi.org/10.1007/3-540-46475-1_16

[201] S. Jiang and X. Zhang, "LIRS," *ACM SIGMETRICS Performance Evaluation Review*, vol. 30, no. 1, p. 31, Jun. 2002. doi: 10.1145/511399.511340. [Online]. Available: http://dx.doi.org/10.1145/511399.511340

[202] L. a. Belady, "A study of replacement algorithms for a virtual-storage computer," *IBM Systems Journal*, vol. 5, no. 2, pp. 78–101, 1966. doi: 10.1147/sj.52.0078. [Online]. Available: http://dx.doi.org/10.1147/sj.52.0078

[203] S. Podlipnig and L. Böszörmenyi, "A survey of Web cache replacement strategies," *ACM Computing Surveys*, vol. 35, no. 4, pp. 374–398, Dec. 2003. doi: 10.1145/954339.954341. [Online]. Available: http://dx.doi.org/10.1145/954339.954341

[204] E. R. Chopra, *Operating System (A Practical Approach)*. New Delhi: S. Chand Limited, 2009. ISBN 9788121931649. [Online]. Available: https://books.google.pt/books?id=aaPIP3rP1A0C

[205] L. A. Belady, R. A. Nelson, and G. S. Shedler, "An anomaly in space-time characteristics of certain programs running in a paging machine," *Communications of the ACM*, vol. 12, no. 6, pp. 349–353, 1969. doi: 10.1145/363011.363155. [Online]. Available: http://dx.doi.org/10.1145/363011.363155

[206] R. Grandl, K. Su, and C. Westphal, "On the Interaction of Adaptive Video Streaming with Content-Centric Networking," in *2013 20th International Packet Video Workshop*. IEEE, Dec. 2013. doi: 10.1109/PV.2013.6691451. ISBN 978-1-4799-2172-0 pp. 1–8. [Online]. Available: http://dx.doi.org/10.1109/PV.2013.6691451

[207] W. Qu, K. Li, H. Shen, Y. Jin, and T. Nanya, "The Cache Replacement Problem for Multimedia Object Caching," in *2005 First International Conference on Semantics, Knowledge and Grid*, no. Skg 2005. IEEE, 2005. doi: 10.1109/SKG.2005.122. ISBN 0769525342 pp. 26–26. [Online]. Available: http://dx.doi.org/10.1109/SKG.2005.122

[208] D. H. Lee, C. Dovrolis, and A. C. Begen, "Caching in HTTP Adaptive Streaming: Friend or Foe?" *Proceedings of Network and Operating System Support on Digital Audio and Video Workshop*, pp. 31:31—-31:36, 2013. doi: 10.1145/2578260.2578270. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2578260.2578270

[209] J. Li, J. Wu, G. Dan, A. Arvidsson, and M. Kihl, "Performance analysis of local caching replacement policies for internet video streaming services," in *2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, vol. 2017. IEEE, Sep. 2014. doi: 10.1109/SOFTCOM.2014.7039112. ISBN 978-9-5329-0052-1 pp. 341–348. [Online]. Available: http://dx.doi.org/10.1109/SOFTCOM.2014.7039112

[210] Yu-Ting Yu, F. Bronzino, R. Fan, C. Westphal, and M. Gerla, "Congestion-aware edge caching for adaptive video streaming in Information-Centric Networks," in *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*. IEEE, Jan. 2015. doi: 10.1109/CCNC.2015.7158039. ISBN 978-1-4799-6390-4 pp. 588–596. [Online]. Available: http://dx.doi.org/10.1109/APCC.2013.6765939

[211] W. Zhang, Y. Wen, Z. Chen, and A. Khisti, "QoE-Driven Cache Management for HTTP Adaptive Bit Rate Streaming Over Wireless Networks," *IEEE Transactions on Multimedia*, vol. 15, no. 6, pp. 1431–1445, Oct. 2013. doi: 10.1109/TMM.2013.2247583. [Online]. Available: http://dx.doi.org/10.1109/TMM.2013.2247583

[212] K. Dong, J. He, and W. Song, "QoE-aware adaptive bitrate video streaming over mobile networks with caching proxy," in *2015 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, Feb. 2015. doi: 10.1109/ICCNC.2015.7069438. ISBN 978-1-4799-6959-3 pp. 737–741. [Online]. Available: http://dx.doi.org/10.1109/ICCNC.2015.7069438

[213] W. Li, S. M. Oteafy, and H. S. Hassanein, "Dynamic adaptive streaming over popularity-driven caching in Information-Centric Networks," in *2015 IEEE International Conference on Communications (ICC)*. IEEE, Jun. 2015. doi: 10.1109/ICC.2015.7249238. ISBN 978-1-4673-6432-4 pp. 5747–5752. [Online]. Available: http://dx.doi.org/10.1109/ICC.2015.7249238

[214] Z. Avramova, D. De , S. Wittevrongel, and H. Bruneel, "Performance analysis of a caching algorithm for a catch-up television service," *Multimedia Systems*, vol. 17, no. 1, pp. 5–18, Aug. 2011. doi: 10.1007/s00530-010-0201-1. [Online]. Available: http://dx.doi.org/10.1007/s00530-010-0201-1

[215] S. Borst, V. Gupta, and A. Walid, "Distributed Caching Algorithms for Content Distribution Networks," in *2010 Proceedings IEEE INFOCOM*. San Diego, CA: IEEE, Mar. 2010. doi: 10.1109/INFCOM.2010.5461964. ISBN 978-1-4244-5836-3. ISSN 0743-166X pp. 1–9. [Online]. Available: http://dx.doi.org/10.1109/INFCOM.2010.5461964

[216] J. Famaey, F. Iterbeke, T. Wauters, and F. De Turck, "Towards a predictive cache replacement strategy for multimedia content," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 219–227, 2013. doi: 10.1016/j.jnca.2012.08.014

[217] H. Abrahamsson and M. Bjorkman, "Simulation of IPTV caching strategies," in *Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2010 International Symposium on*. Ottawa, ON: IEEE, Jul. 2010. ISBN 978-1-56555-340-8 pp. 187–193. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5588896

[218] H. Abrahamsson and M. Nordmark, "Program popularity and viewer behaviour in a large TV-on-demand system," in *Proceedings of the 2012 ACM conference on Internet measurement conference - IMC '12*. New York, New York, USA: ACM Press, 2012. doi: 10.1145/2398776.2398798. ISBN 9781450317054 p. 199. [Online]. Available: http://dx.doi.org/10.1145/2398776.2398798

[219] J. Forlizzi and S. Ford, "The building blocks of experience," in *Proceedings of the conference on Designing interactive systems processes, practices, methods, and techniques - DIS '00*. New York, New York, USA: ACM Press, 2000. doi: 10.1145/347642.347800. ISBN 1581132190 pp. 419–423. [Online]. Available: http://dx.doi.org/10.1145/347642.347800

[220] K. Kilkki, "Quality of Experience in Communications Ecosystem." *J. UCS*, vol. 14, no. 5, pp. 615–624, 2008. [Online]. Available: http://www.jucs.org/jucs_14_5/quality_of_experience_in/jucs_14_05_0615_0624_kilkki.pdf

[221] "Recommendation P.10/G.100: Amendment 2: New Definitions for Inclusion in Recommendation ITU-T P.10/G.100," ITU-T, Tech. Rep. 2006, 2008, Accessed: 2015-09. [Online]. Available: www.itu.int/rec/T-REC-P.10-200807-I!Amd2

[222] M. El-Gendy, a. Bose, and K. Shin, "Evolution of the internet QoS and support for soft real-time applications," *Proceedings of the IEEE*, vol. 91, no. 7, pp. 1086–1104, Jul. 2003. doi: 10.1109/JPROC.2003.814615. [Online]. Available: http://dx.doi.org/10.1109/JPROC.2003.814615

[223] K. De Moor, I. Ketyko, W. Joseph, T. Deryckere, L. De Marez, L. Martens, and G. Verleye, "Proposed Framework for Evaluating Quality of Experience in a Mobile, Testbed-oriented Living Lab Setting," *Mobile Networks and Applications*, vol. 15, no. 3, pp. 378–391, Jan. 2010. doi: 10.1007/s11036-010-0223-0. [Online]. Available: http://dx.doi.org/10.1007/s11036-010-0223-0

[224] "Recommendation BT.500-13: Methodology for the subjective assessment of the quality of television pictures," ITU-R, Tech. Rep., 2012, Accessed: 2015-09. [Online]. Available: http://www.itu.int/rec/R-REC-BT.500-13-201201-I

[225] S. Winkler and P. Mohandas, "The Evolution of Video Quality Measurement: From PSNR to Hybrid Metrics," *IEEE Transactions on Broadcasting*, vol. 54, no. 3, pp. 660–668, Sep. 2008. doi: 10.1109/TBC.2008.2000733. [Online]. Available: http://dx.doi.org/10.1109/TBC.2008.2000733

[226] P. Brooks and B. r. Hestnes, "User measures of quality of experience: why being objective and quantitative is important," *IEEE Network*, vol. 24, no. 2, pp. 8–13, Mar. 2010. doi: 10.1109/MNET.2010.5430138. [Online]. Available: http://dx.doi.org/10.1109/MNET.2010.5430138

[227] R. K. Mok, E. W. Chan, X. Luo, and R. K. Chang, "Inferring the QoE of HTTP video streaming from user-viewing activities," in *Proceedings of the first ACM SIGCOMM workshop on Measurements up the stack - W-MUST '11*. New York, New York, USA: ACM Press, 2011. doi: 10.1145/2018602.2018611. ISBN 9781450308007 p. 31. [Online]. Available: http://dx.doi.org/10.1145/2018602.2018611

[228] T. Porter and X.-H. Peng, "An Objective Approach to Measuring Video Playback Quality in Lossy Networks using TCP," *IEEE Communications Letters*, vol. 15, no. 1, pp. 76–78, Jan. 2011. doi: 10.1109/LCOMM.2010.110310.101642. [Online]. Available: http://dx.doi.org/10.1109/LCOMM.2010.110310.101642

[229] L. Yitong, S. Yun, M. Yinian, L. Jing, L. Qi, Y. Dacheng, and S. Diego, "A study on Quality of Experience for adaptive streaming service," in *2013 IEEE International Conference on Communications Workshops (ICC)*. IEEE, Jun. 2013. doi: 10.1109/ICCW.2013.6649320. ISBN 978-1-4673-5753-1 pp. 682–686. [Online]. Available: http://dx.doi.org/10.1109/ICCW.2013.6649320

[230] 3GPP, "TS 26.247 V12.1.0 Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH) (Release 12)," 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Transparent end-to-end Packet-switched Streaming Service (PSS), Valbonne, Tech. Rep., 2013, Accessed: 2015-09. [Online]. Available: http://www.3gpp.org/DynaReport/26247.htm

[231] K. D. Singh, Y. Hadjadj-Aoul, and G. Rubino, "Quality of experience estimation for adaptive HTTP/TCP video streaming using H.264/AVC," in *2012 IEEE Consumer Communications and Networking Conference (CCNC)*. IEEE, Jan. 2012. doi: 10.1109/CCNC.2012.6181070. ISBN 978-1-4577-2071-0 pp. 127–131. [Online]. Available: http://dx.doi.org/10.1109/CCNC.2012.6181070

[232] S. Mohamed and G. Rubino, "A study of real-time packet video quality using random neural networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 12, pp. 1071–1083, Dec. 2002. doi: 10.1109/TCSVT.2002.806808. [Online]. Available: http://dx.doi.org/10.1109/TCSVT.2002.806808

[233] J. D. Vriendt, D. D. Vleeschauwer, D. Robinson, and B. Labs, "Model for estimating QoE of video delivered using HTTP adaptive streaming," in *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, Ghent, 2013. ISBN 9783901882500 pp. 1288–1293. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6573179

[234] M. Eckert, T. M. Knoll, and F. Schlegel, "Advanced MOS calculation for network based QoE Estimation of TCP streamed Video Services," in *2013, 7th International Conference on Signal Processing and Communication Systems (ICSPCS)*. IEEE, Dec. 2013. doi: 10.1109/ICSPCS.2013.6723923. ISBN 978-1-4799-1319-0 pp. 1–9. [Online]. Available: http://dx.doi.org/10.1109/ICSPCS.2013.6723923

[235] R. Soundararajan and A. C. Bovik, "Video Quality Assessment by Reduced Reference Spatio-Temporal Entropic Differencing," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 4, pp. 684–694, Apr. 2013. doi: 10.1109/TCSVT.2012.2214933. [Online]. Available: http://dx.doi.org/10.1109/TCSVT.2012.2214933

[236] C. Alberti, D. Renzi, C. Timmerer, C. Mueller, S. Lederer, S. Battista, and M. Mattavelli, "Automated QoE evaluation of Dynamic Adaptive Streaming over HTTP," in *2013 Fifth*

International Workshop on Quality of Multimedia Experience (QoMEX). IEEE, Jul. 2013. doi: 10.1109/QoMEX.2013.6603211. ISBN 978-1-4799-0738-0 pp. 58–63. [Online]. Available: http://dx.doi.org/10.1109/QoMEX.2013.6603211

[237] R. K. P. Mok, X. Luo, E. W. W. Chan, and R. K. C. Chang, "QDASH: a QoE-aware DASH system," in *Proceedings of the 3rd Multimedia Systems Conference on - MMSys '12*. New York, New York, USA: ACM Press, 2012. doi: 10.1145/2155555.2155558. ISBN 9781450311311 p. 11. [Online]. Available: http://dx.doi.org/10.1145/2155555.2155558

[238] V. Adzic, H. Kalva, and B. Furht, "Optimizing video encoding for adaptive streaming over HTTP," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 2, pp. 397–403, May 2012. doi: 10.1109/TCE.2012.6227439. [Online]. Available: http://dx.doi.org/10.1109/TCE.2012.6227439

[239] a. E. Essaili, D. Schroeder, D. Staehle, M. Shehada, W. Kellerer, and E. Steinbach, "Quality-of-experience driven adaptive HTTP media delivery," in *2013 IEEE International Conference on Communications (ICC)*. IEEE, Jun. 2013. doi: 10.1109/ICC.2013.6654905. ISBN 978-1-4673-3122-7 pp. 2480–2485. [Online]. Available: http://dx.doi.org/10.1109/ICC.2013.6654905

[240] A. Begen, T. Akgul, and M. Baugher, "Watching Video over the Web: Part 2: Applications, Standardization, and Open Issues," *IEEE Internet Computing*, vol. 15, no. 3, pp. 59–63, May 2011. doi: 10.1109/MIC.2010.156. [Online]. Available: http://dx.doi.org/10.1109/MIC.2010.156

[241] N. Bouten, J. Famaey, S. Latré, R. Huysegems, B. D. Vleeschauwer, W. V. Leekwijck, and F. D. Turck, "QoE Optimization Through In-network Quality Adaptation for HTTP Adaptive Streaming," in *Proceedings of the 8th International Conference on Network and Service Management*. Las Vegas, NV: International Federation for Information Processing, 2012. ISBN 9783901882487 pp. 336–342.

[242] S. Akhshabi, L. Anantakrishnan, A. C. Begen, and C. Dovrolis, "What happens when HTTP adaptive streaming players compete for bandwidth?" in *Proceedings of the 22nd international workshop on Network and Operating System Support for Digital Audio and Video - NOSSDAV '12*. New York, New York, USA: ACM Press, 2012. doi: 10.1145/2229087.2229092. ISBN 9781450314305 p. 9. [Online]. Available: http://dx.doi.org/10.1145/2229087.2229092

[243] R. Houdaille and S. Gouache, "Shaping HTTP adaptive streams for a better user experience," in *Proceedings of the 3rd Multimedia Systems Conference on - MMSys '12*. New York, New York, USA: ACM Press, 2012. doi: 10.1145/2155555.2155557. ISBN 9781450311311 p. 1. [Online]. Available: http://dx.doi.org/10.1145/2155555.2155557

[244] J. Jiang, V. Sekar, and H. Zhang, "Improving Fairness, Efficiency, and Stability in HTTP-Based Adaptive Video Streaming With Festive," *IEEE/ACM Transactions on Networking*, vol. 22, no. 1, pp. 326–340, Feb. 2014. doi: 10.1109/TNET.2013.2291681. [Online]. Available: http://dx.doi.org/10.1109/TNET.2013.2291681

[245] K. Miller, N. Corda, S. Argyropoulos, A. Raake, and A. Wolisz, "Optimal Adaptation Trajectories for Block-Request Adaptive Video Streaming," in *2013 20th International Packet Video Workshop*. IEEE, Dec. 2013. doi: 10.1109/PV.2013.6691452. ISBN 978-1-4799-2172-0 pp. 1–8. [Online]. Available: http://dx.doi.org/10.1109/PV.2013.6691452

[246] K. Miller, E. Quacchio, G. Gennari, and A. Wolisz, "Adaptation algorithm for adaptive streaming over HTTP," in *2012 19th International Packet Video Workshop (PV)*. IEEE, May 2012. doi: 10.1109/PV.2012.6229732. ISBN 978-1-4673-0301-9 pp. 173–178. [Online]. Available: http://dx.doi.org/10.1109/PV.2012.6229732

[247] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2001. ISBN 9781558604896. [Online]. Available: https://books.google.pt/books?id=6hkR_ixby08C

[248] R. Wirth, "CRISP-DM : Towards a Standard Process Model for Data Mining," *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*, pp. 29–39, 2000. doi: 10.1.1.198.5133

[249] G. E. P. Box and D. R. Cox, "An Analysis of Transformations," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 211—-252, 1964. doi: 10.1.1.321.3819. [Online]. Available: http://www.jstor.org/stable/2984418

[250] L. Watthanacheewakul, "A New Family of Transformations for Lifetime Data," in *Proceedings of the World Congress on Engineering 2014*. International Association of Engineers (IAENG), 2014. ISBN 978-988-19252-7-5 pp. 116–121. [Online]. Available: http://www.iaeng.org/publication/WCE2014/WCE2014_pp116-121.pdf

[251] S. Serneels, E. De Nolf, and P. J. Van Espen, "Spatial Sign Preprocessing: A Simple Way To Impart Moderate Robustness to Multivariate Estimators," *Journal of Chemical Information and Modeling*, vol. 46, no. 3, pp. 1402–1409, May 2006. doi: 10.1021/ci050498u. [Online]. Available: http://dx.doi.org/10.1021/ci050498u

[252] J. M. Jerez, I. Molina, P. J. García-Laencina, E. Alba, N. Ribelles, M. Martín, and L. Franco, "Missing data imputation using statistical and machine learning methods in a real breast cancer problem," *Artificial Intelligence in Medicine*, vol. 50, no. 2, pp. 105–115, Oct. 2010. doi: 10.1016/j.artmed.2010.05.002. [Online]. Available: http://dx.doi.org/10.1016/j.artmed.2010.05.002

[253] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, Jul. 2010. doi: 10.1002/wics.101. [Online]. Available: http://dx.doi.org/10.1002/wics.101

[254] P. C. Austin and L. J. Brunner, "Inflation of the type I error rate when a continuous confounding variable is categorized in logistic regression analyses," *Statistics in Medicine*, vol. 23, no. 7, pp. 1159–1178, Apr. 2004. doi: 10.1002/sim.1687. [Online]. Available: http://dx.doi.org/10.1002/sim.1687

[255] R. Bellman and R. E. Bellman, *Adaptive Control Processes: A Guided Tour*, ser. Rand Corporation. Research studies. Princeton University Press, 1961. [Online]. Available: https://books.google.pt/books?id=POAmAAAAMAAJ

[256] M. Kuhn and K. Johnson, *Applied Predictive Modeling*. New York, NY: Springer New York, 2013. ISBN 978-1-4614-6848-6. [Online]. Available: http://dx.doi.org/10.1007/978-1-4614-6849-3

[257] M. Kuhn, "Building Predictive Models in R Using the caret Package," *Journal of Statistical Software*, vol. 28, no. 5, pp. 1–26, 2008. [Online]. Available: http://www.jstatsoft.org/v28/i05

[258] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant Features and the Subset Selection Problem." in *Machine Learning: Proceedings of the Eleventh International Conference*, W. W. Cohen and H. Hirsh, Eds. San Francisco, CA: Morgan Kaufmann, 1994. ISBN 1558603352. ISSN 00189340 pp. 121–129. [Online]. Available: http://machine-learning.martinsewell.com/feature-selection/JohnKohaviPfleger1994.pdf

[259] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes, "Ensemble selection from libraries of models," in *Twenty-first international conference on Machine learning - ICML '04*, vol. 34. New York, New York, USA: ACM Press, 2004. doi: 10.1145/1015330.1015432. ISBN 1581138285 p. 18. [Online]. Available: http://dx.doi.org/10.1145/1015330.1015432

[260] Y. Saeys, I. n. Inza, and P. Larranaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, Oct. 2007. doi: 10.1093/bioinformatics/btm344. [Online]. Available: http://dx.doi.org/10.1093/bioinformatics/btm344

[261] J. Kim, "Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap," *Computational Statistics & Data Analysis*, vol. 53, no. 11, pp. 3735–3745, Sep. 2009. doi: 10.1016/j.csda.2009.04.009. [Online]. Available: http://dx.doi.org/10.1016/j.csda.2009.04.009

[262] A. M. Molinaro, R. Simon, and R. M. Pfeiffer, "Prediction error estimation: a comparison of resampling methods," *Bioinformatics*, vol. 21, no. 15, pp. 3301–3307, Aug. 2005. doi: 10.1093/bioinformatics/bti499. [Online]. Available: http://dx.doi.org/10.1093/bioinformatics/bti499

[263] A. Smola, "Regression Estimation with Support Vector Learning Machines," *Master's thesis, Technische Universit at Munchen*, pp. 1–79, 1996. doi: 10.1.1.10.3628. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.10.3628&rep=rep1&type=pdf

[264] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, Aug. 2004. doi: 10.1023/B:STCO.0000035301.49549.88. [Online]. Available: http://dx.doi.org/10.1023/B:STCO.0000035301.49549.88

[265] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, pp. 318–362, 1986. [Online]. Available: http://dl.acm.org/citation.cfm?id=104293

[266] M. Costa, "Probabilistic Interpretation of Feedforward Network Outputs, with Relationships to Statistical Prediction of Ordinal Quantities," *International Journal of Neural Systems*, vol. 07, no. 05, pp. 627–637, Nov. 1996. doi: 10.1142/S0129065796000610. [Online]. Available: http://dx.doi.org/10.1142/S0129065796000610

[267] P. Domingos, "A Unified Bias-Variance Decomposition and its Applications," in *Proceedings of the Seventeenth International Conference on Machine Learning*. Morgan Kaufmann, 2000. ISBN 2065432969 pp. 231–238. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.30.5038

[268] G. Turner and J. Tay, *Television Studies After TV: Understanding Television in the Post-Broadcast Era*, 1st ed., Routledge, Ed., New York, 2009. ISBN 978-0415477703

[269] D. Tice, "Adventures in Cord-Cutting," 2014, Accessed: 09-2015. [Online]. Available: http://blog.gfk.com/2014/10/adventures-in-cord-cutting/

[270] S. Murray, "OTT to reach nearly half the world's TV households by 2020," Digital TV Research, Tech. Rep., 2014, Accessed: 09-2015. [Online]. Available: https://www.digitaltvresearch.com/ugc/OTTHH2014TOC_toc_105.pdf

[271] M. Proulx and S. Shepatin, *Social TV: how marketers can reach and engage audiences by connecting television to the web, social media, and mobile.* John Wiley & Sons, 2012. ISBN 978-1-118-16746-5

[272] R. Belo, M. Godinho de Matos, and P. Ferreira, "Prime-Time Any Time: The Effect of Time-Shifted TV on Media Consumption," *SSRN Electronic Journal*, pp. 1–17, 2013. doi: 10.2139/ssrn.2242531. [Online]. Available: http://dx.doi.org/10.2139/ssrn.2242531

[273] T. S. Bjøndal and M. Gedde, "Ubiquitous TV: A Business Model Perspective on the Norwegian Television Industry," Master, Norwegian University of Science and Technology, 2011. [Online]. Available: http://brage.bibsys.no/xmlui/handle/11250/266027

[274] Belgacom, "Proximus Q1 Quarterly Report," Belgium, pp. 1–36, 2015, Accessed: 09-2015. [Online]. Available: http://www.proximus.com/sites/default/files/Documents/Investors/Reports/2015/en/Q12015_rapport.pdf

[275] Belgacom, "Proximus Q4 Quarterly Report," Belgium, pp. 1–38, 2014, Accessed: 09-2015. [Online]. Available: http://www.proximus.com/sites/default/files/Documents/Investors/Reports/2014/en/q4/Belgacom_Q4_2014.pdf

[276] Z. Li and G. Simon, "Time-Shifted TV in Content Centric Networks: The Case for Cooperative In-Network Caching," in *2011 IEEE International Conference on Communications (ICC)*. IEEE, Jun. 2011. doi: 10.1109/icc.2011.5963380. ISBN 978-1-61284-232-5. ISSN 05361486 pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/icc.2011.5963380

[277] R. G. Picard, C. H. Davis, F. Papandrea, and S. Park, "Platform proliferation and its implications for domestic content policies," *Telematics and Informatics*, 2015. doi: 10.1016/j.tele.2015.06.018. [Online]. Available: http://dx.doi.org/10.1016/j.tele.2015.06.018

[278] Olswang, "Content meets the cloud: Aereo and the future of cloud TV," pp. 1–19, 2014, Accessed: 09-2015. [Online]. Available: http://www.olswang.com/media/48210818/aereo_report.pdf

[279] United States Court of Appeals for the Second Circuit, "The Cartoon Network LP, LLLP v. CSC Holdings, Inc." pp. 1–44, 2007. [Online]. Available: http://www.ca2.uscourts.gov/decisions/isysquery/339edb6b-4e83-47b5-8caa-4864e5504e8f/1/doc/07-1480-cv_opn.pdf

[280] ThinkTV, "PVRs drive incremental audiences," Mosman, Australia, 2015, Accessed: 09-2015. [Online]. Available: http://www.thinktv.com.au/Media/Stats_&_Graphs/2014/PVRs_drive_incremental_audiences.pdf

[281] J. Moulding, "Swisscom Explains Push For NPVR, As It Heads Towards 50% Time-Shift Viewing," 2014, Accessed: 09-2015. [Online]. Available: http://www.v-net.tv/swisscom-explains-push-for-npvr-as-it-heads-towards-50-time-shift-viewing

[282] Nielsen, "State of the Media: DVR Use in the U.S." pp. 1–6, 2010, Accessed: 09-2015. [Online]. Available: http://www.nielsen.com/content/dam/corporate/us/en/newswire/uploads/2010/12/DVR-State-of-the-Media-Report.pdf

[283] Nielsen, "Time Shift Viewing - Setting the scene for 2012," pp. 1–22, 2011, Accessed: 09-2015. [Online]. Available: http://www.thinktv.co.nz/wp-content/uploads/TSV-Charts2_Part11.pdf

[284] Nielsen, "C3 TV Ratings Show Impact Of DVR Ad Viewing," 2009, Accessed: 09-2015. [Online]. Available: http://www.nielsen.com/us/en/insights/news/2009/c3-tv-ratings-show-impact-of-dvr-ad-viewing.html

[285] Nielsen, "Separating Fact From Fiction," 2016, Accessed: 10-2016. [Online]. Available: http://sites.nielsen.com/newscenter/separating-fact-from-fiction/

[286] M. A. Wahlström and A. Kankainen, "Digital TV Transition and the Hard Disk Drive Revolution in Television Viewing Helsinki Institute for Information Technology HIIT," *International Journal of Communication*, vol. 5, pp. 1606–1622, 2011.

[287] R. Williams and E. Williams, *Television: Technology and Cultural Form*, ser. Routledge classics. Routledge, 2003. ISBN 9780415314565. [Online]. Available: https://books.google.pt/books?id=9XYfPRBR3awC

[288] I. Jennes, J. Pierson, and W. Van den Broeck, "User Empowerment and Audience Commodification in a Commercial Television Context," *The Journal of Media Innovations*, vol. 1, no. 1, pp. 71–87, Feb. 2014. doi: 10.5617/jmi.v1i1.723. [Online]. Available: http://dx.doi.org/10.5617/jmi.v1i1.723

[289] M. Medina, M. Herrero, and C. Etayo, "The impact of digitalization on the strategies of pay TV in Spain," Sociedad Latina de Comunicación Social, La Laguna, Tenerife, Tech. Rep., Apr. 2015. [Online]. Available: http://dx.doi.org/10.4185/RLCS-2015-1045en

[290] D. Mohan, "The evolving value chain in the television industry : changes in pay TV delivery and its implications for the future," Ph.D. dissertation, Massachusetts Institute of Technology, 2014. [Online]. Available: http://hdl.handle.net/1721.1/90718

[291] S. Kalia, "DVR and Its Impact on Indian Market: Now and in Future," *SAGE Open*, vol. 4, no. 4, Dec. 2014. doi: 10.1177/2158244014560551. [Online]. Available: http://dx.doi.org/10.1177/2158244014560551

[292] D. Minoli, *Linear and Non-Linear Video and TV Applications: Using IPv6 and IPv6 Multicast*. John Wiley & Sons, 2012. ISBN 9781118327463 Accessed: 09-2015.

[293] Sling Television, "Sling TV," 2015. [Online]. Available: https://www.sling.com/

[294] Sony, "PlayStation Vue," 2015. [Online]. Available: https://www.playstationnetwork.com/vue/

[295] Simplestream Ltd, "TVPlayer," 2015. [Online]. Available: https://tvplayer.com/

[296] Alcatel-Lucent, "Cloud DVR," 2015, Accessed: 09-2015. [Online]. Available: https://www.alcatel-lucent.com/solutions/cloud-dvr

[297] D. De Vleeschauwer, Z. Avramova, S. Wittevrongel, and H. Bruneel, "Transport capacity for a catch-up television service," in *Proceedings of the seventh european conference on European interactive television conference - EuroITV '09*. New York, New York, USA: ACM Press, 2009. doi: 10.1145/1542084.1542117. ISBN 9781605583402 p. 161. [Online]. Available: http://dx.doi.org/10.1145/1542084.1542117

[298] J. Abreu, V. Becker, and J. Nogueira, "Overview of Catch-up TV and other time-shift TV services," 2015. [Online]. Available: https://13z47x.s.cld.pt

[299] OPTICOM GmbH, "PEVq - Advanced Perceptual Evaluation of Video Quality," 2016, Accessed: 09-2016. [Online]. Available: http://www.pevq.com/

[300] M. Pathan and R. Buyya, "A Taxonomy of CDNs," in *Content Delivery Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, ch. A Taxonomy, pp. 33–77. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-77887-5_2

[301] F. Burden and D. Winkler, "Bayesian Regularization of Neural Networks," in *Artificial Neural Networks*, ser. Methods in Molecular Biology, D. J. Livingstone, Ed. Humana Press, 2009, vol. 458, pp. 23–42. ISBN 978-1-58829-718-1. [Online]. Available: http://dx.doi.org/10.1007/978-1-60327-101-1_3

[302] P. P. Rodriguez and D. Gianola, "brnn (Bayesian regularization for feed-forward neural networks)," 2015, Accessed: 01-2016. [Online]. Available: https://cran.r-project.org/web/packages/brnn/brnn.pdf

[303] A. Liaw, "randomForest," 2015, Accessed: 09-2015. [Online]. Available: https://cran.r-project.org/web/packages/randomForest/randomForest.pdf

[304] B. Ripley and W. Venables, "class," 2015. [Online]. Available: https://cran.r-project.org/web/packages/class/class.pdf

[305] B.-H. Mevik, R. Wehrens, and K. H. Liland, "pls," 2015, Accessed: 09-2015. [Online]. Available: https://cran.r-project.org/web/packages/pls/pls.pdf

[306] A. Karatzoglou, A. Smola, and K. Hornik, "kernlab," 2015, Accessed: 09-2015. [Online]. Available: https://cran.r-project.org/web/packages/kernlab/kernlab.pdf

[307] I. K. Yeo and R. A. Johnson, "A new family of power transformations to improve normality or symmetry," *Biometrika*, vol. 87, no. 4, pp. 954–959, dec 2000. doi: 10.1093/biomet/87.4.954. [Online]. Available: http://dx.doi.org/10.1093/biomet/87.4.954

[308] J. Szlek and A. Mendyk, "fscaret," 2015, Accessed: 09-2015. [Online]. Available: https://cran.r-project.org/web/packages/fscaret/fscaret.pdf

[309] R. Hyndman, "Another Look At Forecast-Accuracy Metrics for Intermittent Demand," *Foresight: The International Journal of Applied Forecasting*, no. 4, pp. 43–46, Jun. 2006, Accessed: 01-2016. [Online]. Available: http://www.researchgate.net/publication/5055536_Another_Look_at_Forecast_Accuracy_Metrics_for_Intermittent_Demand/file/d912f50ff0c2ad9136.pdf

[310] C. Tofallis, "A better measure of relative prediction accuracy for model selection and model estimation," *Journal of the Operational Research Society*, vol. 66, no. 8, pp. 1352–1362, Aug. 2015. doi: 10.1057/jors.2014.103 Accessed: 01-2016. [Online]. Available: http://dx.doi.org/10.1057/jors.2014.103

[311] R. J. Hyndman, "forecast: Forecasting Functions for Time Series and Linear Models," 2015, Accessed: 01-2016. [Online]. Available: https://cran.r-project.org/web/packages/forecast/index.html

[312] R Foundation for Statistical Computing, "The R Project for Statistical Computing," 2016, Accessed: 01-2016. [Online]. Available: https://www.r-project.org/

[313] RStudio Inc., "RStudio," 2016, Accessed: 01-2016. [Online]. Available: https://www.rstudio.com/

[314] HAProxy Technologies, "HAProxy - The Reliable, High Performance TCP/HTTP Load Balancer," 2015, Accessed: 12-2015. [Online]. Available: http://www.haproxy.org/

Appendix A

Time-shift services: a taxonomy and techno-business impacts of Catch-up TV

CENTERIS 2015 - Conference on ENTERprise Information Systems / PRojMAN 2015 - International Conference on Project MANagement / HCIST 2015 - International Conference on Health and Social Care Information Systems and Technologies

Time-shift services: a taxonomy and techno-business impacts of Catch-up TV

Jorge Abreu[a*], Valdecir Becker[b], João Nogueira[a,c], Bernardo Cardoso[c]

[a]*University of Aveiro, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal*
[b]*Federal University of Paraíba, João Pessoa, Brazil*
[c]*Portugal Telecom Inovação, SA, Rua Eng. José Ferreira Pinto Basto, 3810-106, Aveiro, Portugal*

Abstract

This article analyzes the introduction of time-shift TV services, with a focus on Catch-up TV services, and its impact in the Pay-TV market. A taxonomy of these services is proposed in order to contextualize terms like Pause TV, Start-over TV, Personal Video Recorder and Catch-up TV used by Pay-TV providers in their nonlinear TV offerings.

The paper analyzes the techno-business impacts of this technology in the Pay-TV value chain: consumers have more power and can choose when to watch TV shows nonlinearly; service providers have new demands for the technical infrastructure to support the Catch-up TV resources; content providers gain a new way to increase audience.

Despite the challenges brought by Catch-Up TV services to the Pay-TV industry, linear TV will live with nonlinear content in the coming years. Thus, this article offers an updated understanding on ongoing changes in TV market.

Keywords: Catch-up TV; Time-shift; techno-business; taxonomy;

1. Introduction

Television is undergoing a rapid process of changes and transitions[1]. After the introduction of digital TV, still ongoing in many countries[2], new recording features and online videos affect television consumption, production and the industry's business models. From the audience point of view there are a lot of things that are changing how television and other videos are consumed: new larger and thinner screens; multiple devices able to receive signals from broadcast and on-demand; the potential for sharing recorded programs between those devices; the internet of things, which connects all digital devices in the home and on the road; and the audience, which used to be collective and concentrated in the living room, that now happens anywhere, anytime and using any device.

These changes are accelerated by Pay-TV services, which have been established as primary source of access to new television technologies, even in emerging markets, such as Latin America. The cord-cutters phenomenon, where people give up their Pay-TV subscriptions replacing them by OTT services[3,4], requires quick reactions of all links in the television production chain[5,6].

* Corresponding author. Tel.: +351 234 370 200; fax: +351 234 370 868.
E-mail address: jfa@ua.pt.

A significant consequence, noticeable in this scenario, is a change in the way people watch television. Nowadays, clients of advanced Pay-TV systems have multiple and straightforward ways to watch time-shifted TV content, blurring the line between the consumption of linear TV and deferred (previously aired) TV content.

From a user's perspective, it is possible to benefit from several features, enabling them to: pause the linear TV broadcasting and resuming it later; start watching a TV program that is already being broadcast or that they lost; schedule a recording of a TV program or a TV series for later watching; or watch a TV program that aired during the previous 7 days. All of this is possible from the comfort of their couches, using their Smart TVs or their TV sets connected to a Set-Top Box (STB), without dealing with the hassle of connecting other devices to the TV.

After this brief introduction, this paper is structured as follows. Taking into consideration that the existence of a vast range of services supporting time-shifted TV content has been contributing to some misleading interpretation of the associated terminology, in section 2 a taxonomy of the related services is presented, including a functional description of each service, associated user interaction and type of storage involved. The potential techno-business impacts of the Catch-up TV service are addressed in section 3. In this section, the authors address the market motivations that Pay-TV providers should take in consideration. In addition, a description is presented regarding the potential considerations of content providers. Finally, in section 4, the main conclusions and contributions of this paper are presented.

2. A taxonomy of interactive services supporting time-shifted TV content

As the above Pay-TV features are becoming more and more frequent, not only the frontier between linear and time-shifted TV is blurring but also the terminology of the several supporting services is becoming less clear and consistent among the different players of the TV ecosystem. In order to contribute to a clear understanding of all terms involved, a technology-based taxonomy is presented, including: a functional description of the service; the user interaction involved; the type of storage providing the corresponding feature; and other alternative names from which the service is known.

2.1. Pause TV

This is the simplest service, allowing users to pause the television program they are watching - from a few seconds to several minutes or even hours. Users can resume the TV broadcast when they want, continuing to see where they left off; skip a particular segment; or eventually catch up to the linear broadcast[5].

User interaction: To activate this feature, the user only needs to press the "pause" key on the remote control.

Type of storage: [Local Hard Disk (HD) of the STB] - the program is recorded in the local HD, either from the moment the user tunes the respective channel (allowing the user to rewind until the moment the channel was tuned) or only from the moment the user pressed the "pause" key at the remote control. There are however some operators starting to use Network Storage for this feature – this type of storage uses data servers connected through the service provider network, enabling TV Cloud Recordings[7].

Other names: Not applicable (n.a.).

2.2. Start-over TV

In this type of service, users can start watching programs that have already started and, eventually, programs that already finished. The amount of time that is possible to rewind varies from operator to operator ranging from some minutes up to 24 hours.

User interaction: Users have the ability to watch a program, from its beginning or from a prior moment, being this possibility restricted to the tuned channel or offered over other channels (it depends of the type of storage involved - c.f. Type of storage). In the first case, users only need to press the "rewind" key of the remote to go back in time, whereas in the second case they first need to tune to the desired channel. Another possible interaction with the service may be performed by navigating via the Electronic Program Guide (EPG) - channels providing the "Start-over TV" feature are usually marked with a special symbol (e.g. "↶").

Type of storage: [Network Storage or Local Hard Disk of the STB] - the service is usually supported by network storage (in the cloud). Generally, a process of network TV recording ensures that the programs being broadcasted are automatically converted (server side) and stored so they can be made available via the Pay-TV network infrastructure. If the user initiates the "start-over" feature on the tuned channel, the service may be supported by the local HD.

Other names: Restart TV and Time-shift TV - although this is the general expression used when deferred TV contents are at stake, some operators use it in the context of the "Start-over TV" feature.

2.3. PVR

PVR stands for Personal Video Recorder. In this type of service the recordings are subject to the user action, i.e., they only occur if the user proactively schedules a TV program or a series to be recorded, or if he decides to start recording a program that is being watched. The behavior of the service is much the same as the one of a VCR (Video Cassette Recorder), however with a much higher storage capacity and nonlinear access. The user can start watching a performed recording, whenever he wants, even if the program is still being recorded.

User interaction: As mentioned this type of service involves two different type of actions: 1) the user schedules (or initiates) a recording; 2) the user plays one of the recordings. For scheduling the recordings, the user may navigate through the EPG (or eventually make use of an App provided by the operator) or hit the "Rec" key of the remote control to start recording the program being watched. To watch a recorded program, the user needs to go through the menu of the Pay-TV service (or by pressing a shortcut on the remote control) to access the archive of TV recordings.

Type of storage: [Local Hard Disk of the STB or Network Storage] - in its basic format, the service uses the local HD. However, some operators are already using Network Storage for this feature (c.f. section 3).

Other names: DVR (Digital Video Recorder) - it applies when the storage type is local; and nPVR – Network Personal Video Recorder or RS-DVR – Remote Storage Digital Video Recorder - when the storage is in the cloud[8].

2.4. Catch-up TV

This is the most advanced service, relying on an automated process of "live to vod"[9] (offered by companies like Alcatel-Lucent and Fabrix Systems) or on a more restricted process (with editorial control). With this service, TV operators offer recorded content of the last days, on a bouquet up to hundreds of TV channels. The time window of the recordings ranges from a couple of hours up to 30 days, and the number of recorded TV channels varies from operator to operator, according to technical and legal constraints. With this service, users can really, and very easily, catch up TV programs that have been lost or that they explicitly decided to watch later (e.g. watch the news only after they have prepared dinner).

It is worth to notice that despite the broad existence of "Catch-up TV" services accessible via Web (based on portals of some TV channels - like BBC, TV operators or third parties players - like Hulu)[10], the focus of this paper is on "Catch-up TV" via TV. With this approach, we aim to study the technical solution offering a higher impact on the viewers' relation with linear TV, since its usage is remarkably easy and integrated - they do not need to shift to other equipment and screens to watch programs they missed or decided to watch later.

User interaction: As opposed to the usage of a PVR, users do not need to start or schedule recordings, since the Pay-TV operator performs them automatically. Users simply need to "surf the timeline" to watch the automatically recorded programs. They can navigate through the EPG or access the TV recordings archive (generally organized by days and genres).

Type of storage: [Network Storage] - the only type of storage involved relies on a cloud recording infrastructure.

Other names: There are many commercial names eventually with a regional twist (Flashback; Timewarp; Automatic Recordings; Replay; Shift.TV; TV Archive or, e.g. in Spanish Novisto; Te lo perdiste).

3. Potential impacts of an integrated offer of cloud-recorded TV content

Catch-up TV is the reflex of new content-centric paradigms that blur the line between on-demand and linear TV consumption. Because Pay-TV industry is supported on complex relationships between multiple stakeholders (Fig. 1*a)*), the decision of adding a new service must be carefully analyzed in order to consider the established balance of powers between them. The impact of adding a revolutionary service like Catch-up TV to a Pay-TV offering spreads along the complete supply chain, and affects each stakeholder differently.

3.1. Why should Catch-up TV be offered to Pay-TV customers?

The main value proposition of Catch-up TV services lies in consumer empowerment. The control of what to watch, and when, is transferred from the broadcasters to the consumers, and disrupts the established editorial model forcing users to consume whatever is being broadcast at a given time, thus increasing consumer choice.

In a time where cord-cutters[3,4] are becoming a reality, paying attention to the customers and their needs is crucial in order to improve their experience and satisfaction with Pay-TV service providers, hence fostering customer acquisition, retention, and upselling.

Current research[11] indicates that, in Norway, 58% of the consumers were clients of their current Pay-TV service provider for less than 5 years, which indicates that the market is highly dynamic in nature and that users are willing to switch to new service providers in order to take advantage of added features, improved user experience, higher content quality, and lower prices.

In order to determine what features could present an appealing value proposition for customers, a possible approach is to look into the reasons that drive consumers out of the Pay-TV viewing experience into other alternative media services, such as online video. In this regard, ComScore data[12], displayed on Fig. 1 *b)* indicates that two of the main reasons for watching online content are missed TV episodes and the desire to watch past episodes of TV shows. In fact, a study by Accenture[13] indicated that the primary features of interest on Internet-TV are on-demand television viewing and time-shifting. These conclusions suggest that users want to take control on how they watch TV, without being bound to pre-scheduled content, which is exactly what a Catch-up TV service offers. Not giving the consumers a choice will lead to a reduction in Pay-TV service utilization, thus reducing its utility and value from a customer perspective. This reduction in utilization is costly to the Pay-TV providers and to content producers/providers. A positive impact of Catch-up TV on ARPU has been shown in[14].

Broadcasters also benefit from user engagement in Pay-TV services, as the amount of ads watched by users and its cost is much higher than on comparable services, such as on online entertainment sites (Fig. 1 *c), d)*).

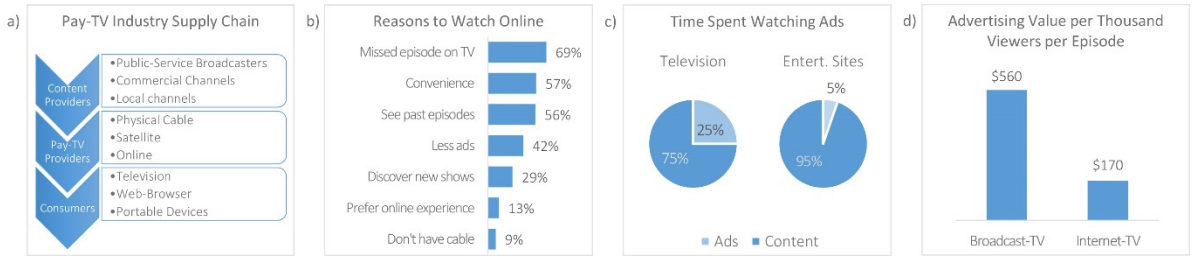


Fig. 1. (a) Summarized Diagram of Pay-TV Industry Supply Chain; (b) Reasons to Watch Video Online[12]; (c) Percentage of Time Spent Watching Ads [21]; (d) The Value of Broadcast vs. Online Viewers[12].

3.2. Impact on Pay-TV Service Providers

While the benefits of the service to the consumers are well established, a Catch-up TV service has a significant impact on service providers' operations and presents several challenges of technical, economic, and legal nature.

Given that Catch-up TV is a service for the masses, with demonstrated large-scale adoption from users[15], it has a high impact on the distribution infrastructures. Unique on-demand viewing sessions need to be established

for each user, hence, traditional broadcasting methods, using multicast on IPTV networks, do not work. The need to unicast video streams imposes large capacity requirements on the networks', which have to be addressed by large investments on networking infrastructure[16].

The fact that Catch-up TV is "data-intensive" is challenging, mostly because the service is usually provided with no added cost, and the users are not charged for the vast amounts of data that needs to transverse the network.

In addition to these technical and economic reasons, there is also the issue of content licensing, given that content providers may impose restrictions on the content available on Catch-up TV and require additional fees. Depending on each country's legislation, and on existing licensing agreements, there may be significant licensing challenges to overcome in order to add TV stations to the Catch-up TV lineup.

In countries where Pay-TV Service Providers offer a wide range of channels in Catch-up TV, they do it on the premise that it is a kind of nPVR service, where the customer schedules full channels for recording instead of just some shows. This seems valid for countries where shared copy for NPVR is allowed and no additional compensation is due to the content owners. In countries like Germany, France and the USA, where law mandates private copy, even on nPVR services[17,18], this full lineup for Catch-up TV services does not exist, instead, the only channels available in this format have a licensing agreement with the service provider, or channels that themselves have a Catch-up TV service in place, like most free-to-air channels in France or in the UK.

3.3. Impact on Content Providers

Content providers decide the price of the content and, therefore, have a high bargaining power in the supply chain, which is used to leverage new forms of delivery as an opportunity to increase revenue, such as demanding micro-payments (pay-per-view), or charging additional subscription fees to authorize different delivery services.

However, limiting the availability of content to Catch-up TV services may be counter-productive. Depending on the individual TV station business model, the reasons may vary. For premium TV stations, were the advertising revenue is residual and most revenue is from user subscriptions, not allowing a service like Catch-up TV reduces its value proposition for the consumers. This is especially true if the aired content does not have any temporal relevance, which is usually the case of movies and TV series premium channels, but also applies to sports channels, or other TV stations where live events are particularly relevant.

Ultimately, because Catch-up TV increases overall media consumption, the content providers get an increased exposure of their programs, and advertisements, to consumers. This motivated Nielsen, in the USA, to release the so-called "C3" ratings that measures commercials watched both live and in a 3 days window in DVR, where it is shown that, although it has very little impact on live events, like sports competitions, it represents a big boost on serialized TV shows, sometimes more than a full rating point[19].

4. Conclusions

This paper investigated the impact of time-shift technologies on the Pay-TV market. To identify the services, the differences and resources scope, a taxonomy was proposed. Among the services analyzed, "Catch-up TV" is the most advanced service in the field, presenting a remarkable potential for changing viewers' relation with linear TV, leveraging a non-linear experience. Moreover, the article presents potential techno-business impacts on the Pay-TV value chain.

This research brings new elements for analyzing the changing process in television ecosystem. For many years, people just watched on TV what was broadcasting. In many countries, some free-to-air channels monopolized the audience. The quality of programming was something relegated to the background, as there was little to no choice[20]. Nowadays, the expansion on the number of available TV channels, leveraged by Catch-Up TV services, provides users with a much larger choice of programs, namely those already broadcasted.

From a business perspective, preventing the cord-cutting phenomenon, reducing churn, and increasing the ARPU (Average Revenue Per User), is essential and can only be achieved by providing a rich and attractive service offering empowered by Catch-up TV services.

From a technological point of view, broadcasters could offer all programs simultaneously in the cloud, and the viewer could choose what and when to watch, regardless of the transmission time, from a much larger TV content offer. That is, with the new technological resources, content quality becomes the differentiation factor, and not the lack of choice or transmission time.

Acknowledgements

The authors are grateful to IAPMEI, QREN and COMPETE for funding the GAPOTT project (2013/34009).

References

1. Turner G, Tay J. Television Studies After TV: Understanding Television in the Post-Broadcast Era. 1st ed. Routledge, editor. New York; 2009. 224 p.
2. Sistema Brasileira de TV Digital [Internet]. [cited 2015 Apr 1]. Available from: http://forumsbtvd.org.br/
3. Tice D. Adventures in Cord-Cutting [Internet]. 2014 [cited 2015 Mar 24]. Available from: http://blog.gfk.com/2014/10/adventures-in-cord-cutting/
4. Murray S. OTT to reach nearly half the world's TV households by 2020. Digit TV Res [Internet]. 2014 [cited 2015 Mar 26];(November). Available from: https://www.digitaltvresearch.com/ugc/OTT HH 2014 TOC_toc_105.pdf
5. Definition of linear TV [Internet]. [cited 2015 Mar 26]. Available from: http://www.itvdictionary.com/definitions/linear_tv_definition.html
6. Proulx M, Shepatin S. Social TV: how marketers can reach and engage audiences by connecting television to the web, social media, and mobile. John Wiley & Sons; 2012.
7. Noam E. Cloud TV: Toward the next generation of network policy debates. Telecomm Policy [Internet]. 2014;38(8-9):684–92. Available from: http://www.sciencedirect.com/science/article/pii/S0308596113001766
8. Olswang. Content meets the cloud: What is the legality of cloud TV [Internet]. 2013 [cited 2015 Mar 26]. Available from: http://www.olswang.com/media/34009878/go_230_cloudpvr_v4_lo-res.pdf
9. Elemental. Fast Forward: Implementing Live-to-VOD Services [Internet]. 2015 [cited 2015 Mar 20]. Available from: http://www.digitaltveurope.net/309572/fast-forward-implementing-live-to-vod-services/
10. Video on demand and catch-up tv in europe [Internet]. European Audiovisual Observatory. [cited 2015 Mar 26]. Available from: http://www.obs.coe.int/documents/205595/264625/VOD+2009+EN.pdf/78bbecb7-7c8f-4b67-8771-1189872a9637
11. Bjøndal TS, Gedde M. Ubiquitous TV: A Business Model Perspective on the Norwegian Television Industry [Internet]. Norwegian University of Science and Technology; 2011. Available from: http://brage.bibsys.no/xmlui/handle/11250/266027
12. Piech D. The State Of Online Video. comScore [Internet]. 2010;1–41. Available from: http://www.comscore.com/content/download/7235/125253/version/1/file/comScore+OMMA+Video+Presentation+-+Jan+2011.pdf
13. Venturini F. The future of broadcasting: Sustaining shareholder value and high performance in a changing industry. Accenture [Internet]. 2008; Available from: http://www.accenture.com/SiteCollectionDocuments/PDF/FutureBroadcastingFinalSingle.pdf
14. Belo R, Matos MG De, Ferreira P. Prime-Time Any Time : The Effect of Time-shifted TV on Media. TPRC41 Research Conference on Communication, Information and Internet Security September 27-29. 2013. p. 1–17.
15. CNC. L ' économie de la télévision de rattrapage en 2014. Cent Natl du cinéma l'image animée [Internet]. 2015;1–33. Available from: http://www.cnc.fr/web/fr/ressources/-/ressources/6592632
16. Li Z, Simon G. Time-shifted TV in content centric networks: The case for cooperative in-network caching. IEEE International Conference on Communications. 2011.
17. The Cartoon Network LP, LLLP v. CSC Holdings, Inc. United States Court Appeals Second Circuit [Internet]. 2007;1–44. Available from: http://www.ca2.uscourts.gov/decisions/isysquery/339edb6b-4e83-47b5-8caa-4864e5504e8f/1/doc/07-1480-cv_opn.pdf
18. Olswang. Content meets the cloud: Aereo and the future of cloud TV [Internet]. 2014 [cited 2015 Apr 1]. p. 1–19. Available from: http://www.olswang.com/media/48210818/aereo_report.pdf
19. Nielsen. C3 TV Ratings Show Impact Of DVR Ad Viewing [Internet]. 2009 [cited 2015 Mar 31]. Available from: http://www.nielsen.com/us/en/insights/news/2009/c3-tv-ratings-show-impact-of-dvr-ad-viewing.html
20. Arlindo Machado. A televisão levada a sério. São Paulo: Senac; 244 p.
21. IBM. Beyond content: Capitalizing on the new revenue opportunities. Inst Bus Value [Internet]. 2009;1–20. Available from: http://public.dhe.ibm.com/common/ssi/ecm/gb/en/gbe03361usen/GBE03361USEN.PDF

Appendix B

Survey of Catch-up TV and Other Time-Shift Services: A Comprehensive Analysis and Taxonomy of Linear and Nonlinear Television

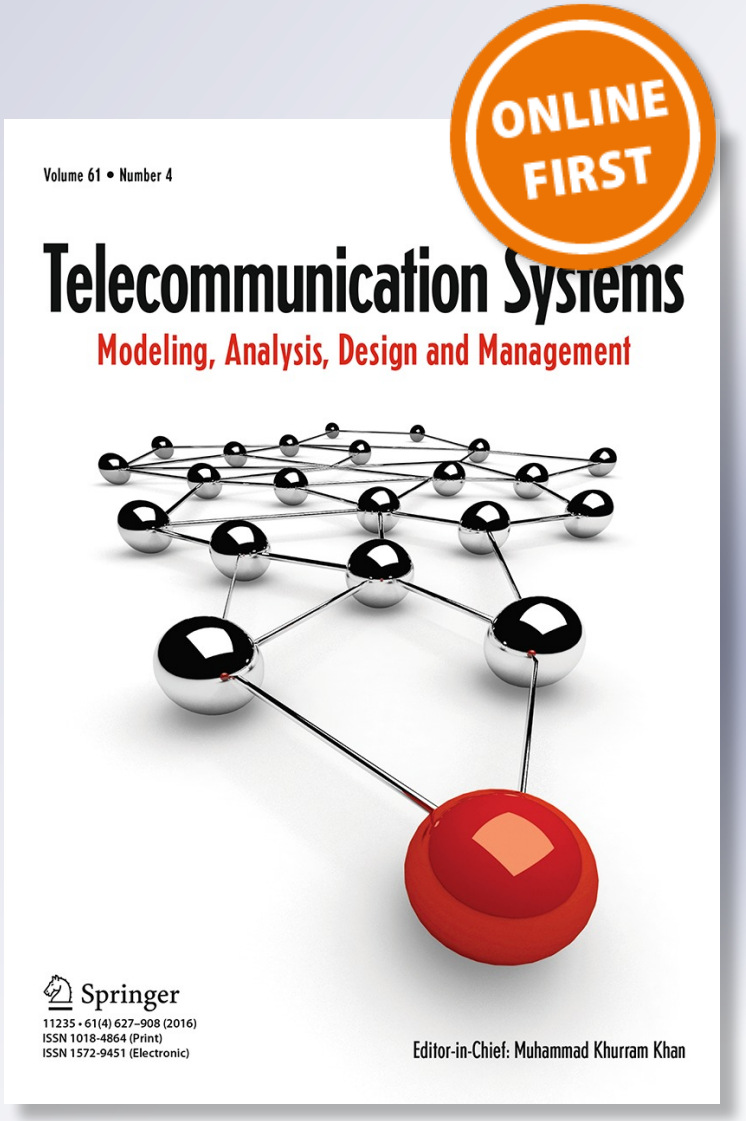
Survey of Catch-up TV and other time-shift services: a comprehensive analysis and taxonomy of linear and nonlinear television

Jorge Abreu, João Nogueira, Valdecir Becker & Bernardo Cardoso

Telecommunication Systems
Modelling, Analysis, Design and Management

ISSN 1018-4864

Telecommun Syst
DOI 10.1007/s11235-016-0157-3



Springer

Telecommun Syst
DOI 10.1007/s11235-016-0157-3

Survey of Catch-up TV and other time-shift services: a comprehensive analysis and taxonomy of linear and nonlinear television

Jorge Abreu[1] · **João Nogueira**[1,2,3] · **Valdecir Becker**[4] · **Bernardo Cardoso**[2]

Abstract This study analyzes recent changes on the television (TV) market, and the transformation of TV consumption habits, motivated by new transmission and recording technologies. Through an international survey spanning 62 countries and 4 continents, a growing presence of nonlinear TV services and over-the-top (OTT) content offerings is found. A detailed taxonomy of ways of watching TV content on the TV set is proposed to clarify the existing Pay-TV and OTT services, including time-shift and video-on-demand (VoD) terms such as Pause TV, Start-over TV, personal video recorder, Catch-up TV, transaction VoD, and subscription VoD, to name a few. An in-depth literature review is conducted, focusing not only on the technological aspects of nonlinear TV, but also on its business and consumer behavior impacts. The existing research works clearly demonstrate a lack of global reach by mostly concentrating their analyses on specific regions or countries. In addition to the literature review, the survey and taxonomy definition, the impact of nonlinear TV on the complete supply chain is also extensively discussed, indicating structural changes not yet addressed in the current literature.

✉ João Nogueira
joaonogueira@ua.pt

Jorge Abreu
jfa@ua.pt

Valdecir Becker
valdecir@ci.ufpb.br

Bernardo Cardoso
bernardo@telecom.pt

[1] University of Aveiro, Aveiro, Portugal

[2] Altice Labs, SA, Aveiro, Portugal

[3] Instituto de Telecomunicações, Aveiro, Portugal

[4] Federal University of Paraíba, João Pessoa, Brazil

Keywords Time-shift · Television · Taxonomy · Survey · Catch-up TV · Over-the-top

1 Introduction

The scientific interest in television (TV) is quite broad. It has been the focus of various research topics, including the development of production, transmission and reception technologies, their influence on how people watch TV, on marketing and advertising, and also on the political, psychological, and sociological impacts of TV programs, to name a few.

Historically, TV is considered an organized and planned one-way communication media, where each programming unit transitions smoothly to the next. The most important characteristics of TV are:

1. Content is organized in channels, which seek identification and recognition by viewers. Each channel has a name, a visual identity and a programming profile, formulated with the purpose of keeping the viewers watching;
2. Channels organize content in individual units, the TV programs and commercials. This organization aims to hold the viewer's attention by interleaving different programming profiles;
3. Programs are promoted and presented to viewers extensively through internal-promotions [6]. A key feature of these self-promotions is the positive self-reference, where the next episode or program is promised to be better than the last one, thus sustaining the "eternal to come" adage [70].

These characteristics are affected by digital recording features, through digital video recorders (DVRs) or other

Published online: 23 March 2016

services, as the traditional TV flow ceases to exist when programs become available independently.

In these modern scenarios, viewers are able to define the organization of TV programming by choosing the TV station and the program they are interested in.

It is well established that technology plays a crucial role in the TV usage and value generation in the broadcasting market [70], thus, a great deal of attention has been given to the development and introduction of new technologies [14,45], to changes in audience behavior [7,29], and to impacts on market and business models [38,41]. Studies have been conducted on how content recording impacts different countries [33,38], however, even though some studies reported an increased usage of Catch-up TV, there has been little research on the international market and scientific community on how to organize and classify these new services. TV analysis is usually focused on local research, with limited implications and conclusions regarding international offerings of Catch-up TV, VoD and over-the-top (OTT) services. Moreover, to the best of the authors' knowledge, there is no taxonomy of those services, which is an essential starting point for a universal, international, analysis.

This study examines Catch-up TV and other nonlinear services in 62 countries, spread across four continents, to identify and quantify their availability on managed operator networks (MONs), and shows that the nonlinear services are becoming ubiquitous.

When offered through the TV set, Catch-up TV provides a significant contribution to a great user experience. This unique characteristic paves the way for a remarkable worldwide penetration, as demonstrated by the fact that the first commercial releases have no more than 8 years, and already represent a first class feature in a very significant number of countries—74 operators from 34 countries provide it.

To fully understand the on-demand scenario to which the TV heads towards, a taxonomy of linear and nonlinear TV services operated by managed network operators and OTT providers is presented.

This work indicates a reorganization of the TV market, with audiences migrating from linear to nonlinear TV. A strengthening of live TV is also identified, with content creators seeking interesting programs, that are worth being watched during live broadcast instead of recorded and watched at a later time.

In addition to framing the business relevance of Catch-up TV and the importance of cloud-based time-shifting solutions, this study also addresses the technological implications of these services.

This paper is organized as follows. An initial state-of-the-art review is presented, summarizing the most important research identified in recent years regarding Catch-up TV and new technologies introduced in the TV ecosystem. Next, a complete taxonomy is proposed and discussed. A worldwide overview of services offering nonlinear TV content over MONs ensues, and the impacts of the increased availability of Catch-up TV services are analyzed. The final section presents the most relevant conclusions.

2 Literature review

Several studies indicate a revolution in the TV ecosystem due to the introduction of manual and automatic recordings, recommendation and retrieval technologies for TV content. As a consequence of the widespread usage of these technologies, an objective analysis of the content enjoyment and changes in TV audience habits is finally possible. In order to frame the relevance of this paper an in-depth review of the literature is performed focusing on nonlinear TV viewing practices and their impact on the TV ecosystem. The related research is concentrated on three prominent themes: the development and introduction of new technologies, changes in audience behavior, and impact on market and business models.

From a technological standpoint, one of the first consequences of nonlinear TV services is the increase on networks' unicast traffic. To lessen this problem, [28] and [30] suggest the use of decentralized storage whenever possible. This approach was studied in [45] for cable TV networks, and in [25,37,68] for IP television (IPTV) services.

Other authors [14] developed algorithms to predict which content is most sought after, and are able to decide if and where each program should be recorded—popular content is recorded in the set-top-boxes (STBs)' hard disk drives (HDDs), while less popular content is stored in the cloud. These technologies have a significant impact on network and infrastructure management. However, it is necessary to stress that, for the viewer, the most important aspects are service usability, content availability, and cost.

From a behavioral perspective, [14] presents a descriptive and inferential statistical analysis on viewing practices (time-shifted, online and mobile), based on data collected over a six-month period in 2010–2011. Regarding the popular time-shift services, the authors consider that they do not alter the traditional conceptualization of TV as a broadcast medium; however, they do not make a clear differentiation between the diverse time-shift services (Pause-TV, Start-over TV, PVR and Catch-up TV) as we do in the taxonomy section. Online viewing, considered an emerging mode that blurs the boundary between TV and new media, is seen by the authors as comprising peer-to-peer (P2P), Bit-Torrent and video streaming from network TV station sites or dedicated services (e.g. Netflix). As for the motivation that drives respondents to watch content on their computers instead of their TV sets, the reason that stands out is the lack of content availability on broadcast TV (42.5 %). Finally, they present

mobile viewing, mostly through dedicated applications, as the most recent consequence of digital convergence. Despite the potential evolutions registered from 2010 to 2011 until now, the paper gives worthy insights about the differences across three key demographic variables: gender, age, and region of residence.

One of the most recent papers is [66], which performs an interesting comparison of broadcast TV viewing behaviors with several nonlinear services (Catch-up TV, VoD streaming services, content recording and downloading). They found that TV series and movies are mostly watched through nonlinear services, and also corroborated that people's attention to content is more focused when nonlinear services are at stake, whereas with regular broadcasts (news, talk shows and other "lighter" TV genres) the adoption of multitasking behavior is more frequent. Finally, the authors also illustrate that the hassle of dealing with the several fragmented services, with different qualities, prices, and technological issues can make it hard for users to watch TV the way they want.

This merging of household media devices and delivery systems was already pointed by Jenkins when he referred to the Black Box Fallacy [28].

These works are consistent with other research, such as [7], which claims that online content consumption is more concentrated in time and quantity than offline viewing, contradicting the hypothesis of a long tail effect of Catch-up TV. The authors state that 69 % of the videos have the same success online and offline; 16 % of the videos are not successful in any platform and only 15 % benefit from being available online. The temporality of replay TV consumption is very close to live broadcasting, thus softening rather than breaking the synchrony of traditional TV. The largest consumption of online videos happens in the first 3 days of their appearing, with 58 % of the total views. Similar results were attained by [44], which adds that users overwhelmingly prefer serialized content.

Additional studies analyze how users can gain more control or power (empowerment) over their media experiences [29], the different ways of watching TV [26,63], and how behaviors change according to audience profiles [4,16,56]. Finally, [67] considers the introduction of content recording technologies as a natural evolution that substantially changes the way people watch TV.

Mohan [41]'s business-model research points to a need for new measurement technologies for online video, to a continued pressure against bundling, the upward integration in terms of industry activities, the potential downward OTT offerings by the major networks, and the risk of avalanche decline in cable subscribers.

Similarly, [33,38] studied the impacts of digitization and DVR usage in the TV market, concluding that TV viewing experience has been completely revolutionized with the advent of digital technologies, with market and business impacts similar to those identified in [41]. Wirtz [71] conducts a more comprehensive analysis, focused on an integrated management perspective on business models, value chains and competencies. According to the authors, these three approaches are complementary with regard to the higher goals of generating competitive advantages in media companies.

Despite the scientific potential and impact of existing research, there are gaps that the present work addresses, including a worldwide survey comprising 62 countries in 4 continents, and the identification and quantification of services offering nonlinear TV content over MONs. Another tackled gap is related to the lack of a suitable taxonomy, or any other type of classification, concerning the existing modes of watching TV programs. With these contributions, a new insight on the impact of modern TV consumption paradigms is provided that not only addresses the understanding of consumer demand for new services, but also fosters new approaches that strengthen the appeal of linear TV.

3 A taxonomy of ways of watching TV content over the TV set

The frontier between linear TV and other forms of watching TV content is blurring, as is the corresponding terminology, which is becoming less clear and consistent not only among the different players of the TV ecosystem but also within academics, as previously observed.

In order to provide a clear understanding of the ways of watching TV content on the big screen, Fig. 1 depicts a matrix with four quadrants. The columns separate linear and nonlinear content, while the rows distinguish managed and unmanaged OTT network delivery.

This macro organization is not completely hermetic, since some services that are put into a particular quadrant might be found on neighbor sectors, although with a minor expression. Regarding transmission types, the focus is on legal broadcast and streaming offers, and do not contemplate download & play content.

3.1 Linear content over managed operator networks (Q1)

Linear TV, i.e. "regular TV broadcast" respecting a predetermined program lineup [40], was considered for decades as the traditional and more popular way of watching TV programs.

This is still the dominant way of watching TV from national free-to-air TV services and major Pay-TV Operators such as British Telecom in England, NET in Brazil, Time-Warner in the USA and MEO in Portugal, however,

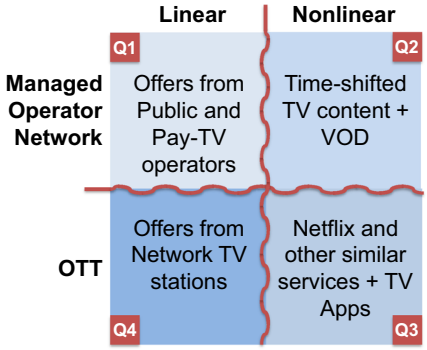


Fig. 1 Four major quadrants of ways of watching TV

customers are moving to other quadrants as detailed in the forthcoming sections.

3.2 Nonlinear content over managed operator networks (Q2)

With the advent of interactive services supporting deferred TV content, e.g. Catch-up TV, major Pay-TV operators started offering time-shifted TV in addition to VoD content through user-friendly TV interfaces.

Unlike the straightforward classification of services belonging to Q1, both time-shifted TV and VoD deserve a more detailed explanation. Therefore,a technology-based taxonomy is provided including a functional description of these services, the user interaction involved, the type of storage supporting each feature—if based on a local hard disk drive (HDD) or in the cloud [50]-, and the other names by which the services are known.

3.2.1 Time-shift TV

Time-shift TV refers to the visualization of deferred TV content, i.e. linear-TV content that is recorded to be watched later, using one of the following services.

Pause TV Users may pause the program they are currently watching, from a few seconds up to several hours. They can resume the TV broadcast when they want, continuing where they left off, skip a particular segment, or catch up to the linear broadcast.

- *Interaction* To use the feature, users must press the "play/pause" key on their remote control.
- *Storage* HDD of the STB or Network Storage. The program is recorded either from channel tune time (allowing the user to rewind until then) or from the moment when the user pressed the "pause" key at the remote control.

Start-over TV In this service, users can watch programs that have already started or finished from the beginning. The amount of time that is possible to rewind varies from operator to operator—from some minutes up to 24 h—and so does the number of TV channels supporting this feature.

- *Interaction* Users may watch a program from its beginning or other moment. This possibility may be restricted to the tuned channel or offered on other not-yet-tuned channels, depending on the type of storage involved. In the first case, they can only rewind up to the point in time when they last tuned the channel, whereas on the second they may press the "rewind" key of the remote control to go back in time up to an operator-configured maximum time window, or use the electronic program guide (EPG) to select a program to restart.
- *Storage* Network storage or hard disk of the STB. The service is usually supported by network/cloud storage. If the user starts the feature on the tuned channel, the service may be supported by the local Hard Disk Drive (HDD).
- *Other names* Restart TV and time-shift TV - although this is the general expression used when deferred TV contents are at stake, some operators use it in the context of the "Start-over TV" feature.

Personal video recorder (PVR) In this case, the recordings depend on user action, i.e. they only occur if the user proactively schedules a TV series or program recording, or if he decides to start recording a program that is being watched. Its behavior is similar to that of a video cassette recorder (VCR), however with a larger storage capacity and nonlinear access. The user can start watching a recording when he wants, even if the program is still being recorded.

- *Interaction* To schedule recordings, the user may navigate the EPG, make use of an external application, or hit the "Rec" key of the remote control to start recording the program currently being watched. To watch a recording, the user needs to use the service interface, or press a dedicated button on the remote control to access the archive.
- *Storage* Hard disk of the STB or network storage. In its basic format, the service uses the local HDD. Some operators rely on Network Storage for this feature.
- *Other names* DVR—applies when the storage type is local, i.e. in the STB's HDD; and network personal video recorder (NPVR), or remote storage digital video recorder (RS-DVR), when the storage is in the cloud [51].

Catch-up TV is the most advanced service, relying either on an automated process of "Live to VoD" [22], or on a

more restricted editorial-control process. TV operators offer recorded content of the previous days, on up to hundreds of TV stations. The time window of the recordings ranges from a couple of hours up to 30 days, and the number of recorded TV stations varies from operator to operator, according to technical, legal, and business constraints. Using this service, users can catch up on TV programs that have been missed or that they explicitly decided to watch.

It is worth to notice that despite the broad existence of *Catch-up TV* services accessible via Web, based on portals of some TV channels (e.g. BBC, TV operators, or third parties players such as Hulu) [34], the focus of this taxonomy is on ways of watching TV using the TV set. This approach aims at studying the technical solutions with a high impact on the viewers' relationship with linear TV, since its usage is remarkably easy and integrated, as they do not need to shift to other devices.

- *Interaction* As opposed to PVRs, users do not need to schedule recordings, since the Pay-TV operator performs them automatically. They simply need to "surf the timeline" to watch the automatically recorded programs, navigate through the EPG, or access the TV recordings archive (generally organized by days and genres) through the menu or by pressing a dedicated key on the remote.
- *Storage* Network Storage, relying on a cloud-recording infrastructure.
- *Other names* There are many commercial names usually with a regional twist (Flashback; Timewarp; Automatic Recordings; Replay; Shift.TV; TV Archive or, e.g. in Spanish Novisto; Te lo perdiste).

3.2.2 Video-on-demand (VoD)

When referring to VoD the only considered services are those where users need to pay to watch a specific TV content using one of the following options.

Transaction VoD (T-VoD) is the most typical version of the service, where customers need to pay a given amount of money whenever they want to watch a content from the catalog. The rental time is usually of 24 or 48 h, during which they can watch it several times.

Electronic sell through VoD (EST-VoD) is a version of the VoD service involving the payment of a one-time fee to access the purchased content without restrictions, usually on a specific platform [37]. Although this type of VoD is more frequent on OTT providers like Apple iTunes and Amazon Instant Video, it also being offered by traditional Pay-TV operators, like Verizon's FiOS TV [25].

Subscription VoD (S-VoD) refers to the business model whereby customers pay a monthly fee to watch whatever they want from the provider catalog for an unlimited number of times. Like the EST-VoD version, it is no longer an exclusive option of these providers, since Pay-TV operators are also offering S-VoD. An example is the Disney VoD service offered by several Pay-TV operators like AT&T, Cablevision or Comcast [68].

3.3 Nonlinear content via OTT (Q3)

Q3 shifts from nonlinear TV content offered by Pay-TV operators to content (mostly movies and series) delivered over the Internet without the involvement of Pay-TV operators.

Bridging devices, such as computers, smartphones or tablets, are central to this scenario, and to some extent to the linear scenario of quadrant Q4 as well, by letting users watch TV on the big screen.

3.3.1 Over-the-top (OTT) providers

When looking into OTT providers, Netflix, Amazon Instant Video, and Hulu are some of the names that ring the bell, but there are many other OTT providers that are sidestepping operator participation and control, including Apple, Sony and Dish Push [55].

According to [31], consumer adoption of these services is surging, driven by the increase in broadband Internet access and the availability of the services in a multi-platform approach simplified by "bridging devices", that are making the process of watching OTT content using personal computers (PCs), gaming consoles, smartphones and tablets on the big screen more straightforward and accessible.

[31] predicts that subscribers of services like Netflix and Amazon Prime Instant Video will grow from 92.1 million in 2014 to 333.2 million by 2019. This massive adoption is a real threat for Pay-TV operators since these subscribers are potential cord-cutters, no longer interested in expensive Pay-TV offers.

3.3.2 Dedicated apps

TV broadcasters are aware of this trend, and started offering dedicated applications for watching nonlinear TV content over the Internet. Applications from BBC, CBS, Fox, History, and NBC are examples of this initiative.

Several Pay-TV operators also offer their own applications, letting customers access their TV subscriptions, NPVR, Catch-up TV and VoD content. A few examples include Comcast XFINITY TV GO [18], Time Warner Cable TV [62] or MEO GO [39].

Table 1 Countries and operators offering Catch-up TV and other time-shift TV services

	# Countries analyzed			# Operators with Catch-up TV and other nonlinear TV services							
	Total	With Catch-up TV	Without	Catch-up	Pause	Start-Over	NPVR	DVR	T-VoD	EST-VoD	S-VoD
Europe	30	19	11	37	37	34	6	29	34	3	15
America	15	7	8	23	23	23	3	23	20	0	10
Asia	15	6	9	12	12	7	1	8	12	0	1
Oceania	2	2	0	2	2	0	0	2	2	0	0
Total	62	34	28	74	74	64	10	62	68	3	26

3.4 Linear content via OTT (Q4)

In this quadrant, there are different approaches for watching linear-TV over the Internet, i.e. in an OTT way.Traditional broadcasters and Pay-TV operators tend to offer web sites, dedicated applications and players, while recent competitors provide pure-OTT alternatives such as Sling TV [58], or PlayStation Vue [59].

In the latter case, these offers are independent from any Pay-TV operator, their customers are real cord-cutters, relying only on an internet service provider (ISP) contract for watching linear-TV for a free or small monthly fee. TVPlayer [57] is an example.

4 A worldwide overview of services offering nonlinear TV content over managed operator networks

Considering this paper's focus on the exploration of new viewing practices of nonlinear TV over MONs, a survey is performed on the worldwide offer of services belonging to quadrant Q2 of the taxonomy proposed in the previous section (Fig. 1).

The dominant potential of Catch-up TV services and their impact on the TV ecosystem is presented in this quadrant. When offered through the TV set, Catch-up TV has a strong impact on the TV ecosystem, and significantly contributes to a great user experience, as demonstrated by its worldwide penetration growth since 2007 [5,20].

When mapping the worldwide existence of Pay-TV operators offering Catch-up TV services, the opportunity is taken to report on additional nonlinear TV services: Pause-TV, Start-over TV, PVR and VoD.

4.1 Data gathering methodology

A systematic methodology is employed to perform a thorough overview of Pay-TV operators in Europe, America, Asia and Oceania supporting nonlinear TV services. Using a worldwide list of Pay-TV operators [69], the web-sites of major providers from 62 countries are visited. When applicable, Google's automatic translation tool is used.

Due to interest on the current footprint of Catch-up TV services, operators offering the service are listed in a spreadsheet with the following key fields: country; operator; Catch-up TV product name; and time window of previously aired programs.

As for Catch-up TV details, following time windows are considered: up to 3 days; between 3 and 7 days; more than 7 days; and "other" when the time span depends on independent broadcaster agreements. In addition to Catch-up TV data, the spreadsheet available in [4] includes other time-shift TV services provided by the operators at stake: Pause-TV, Start-over TV and PVR supported on the local HDD (DVR) and on cloud based storage (NPVR). The availability of T-VoD, EST-VoD, and S-VoD services is also reported.

5 Results

As shown in Table 1, from the 62 countries analyzed, 34 have one or more operators offering Catch-up TV services, whereas in the remaining 28 countries they are absent, to best of authors' knowledge.

Other time-shift services that those operators are providing are also listed. It is possible to observe that most operators with Catch-Up TV services also offer Pause-TV, Start-over TV, DVR and T-VoD, while NPVR, EST-VoD and S-VoD are less widespread. The infographic of Fig. 2 was produced using this data.

One can observe that American countries (especially in the North), European countries (mainly in the West) and others in Asia and Oceania have a significant presence of operators offering Catch-up TV and additional nonlinear TV services. As the implementation of these services is market oriented, their footprint reflects a significant demand for nonlinear TV. This represents a thrilling evolution of the actual TV ecosystem, and an exciting research field with

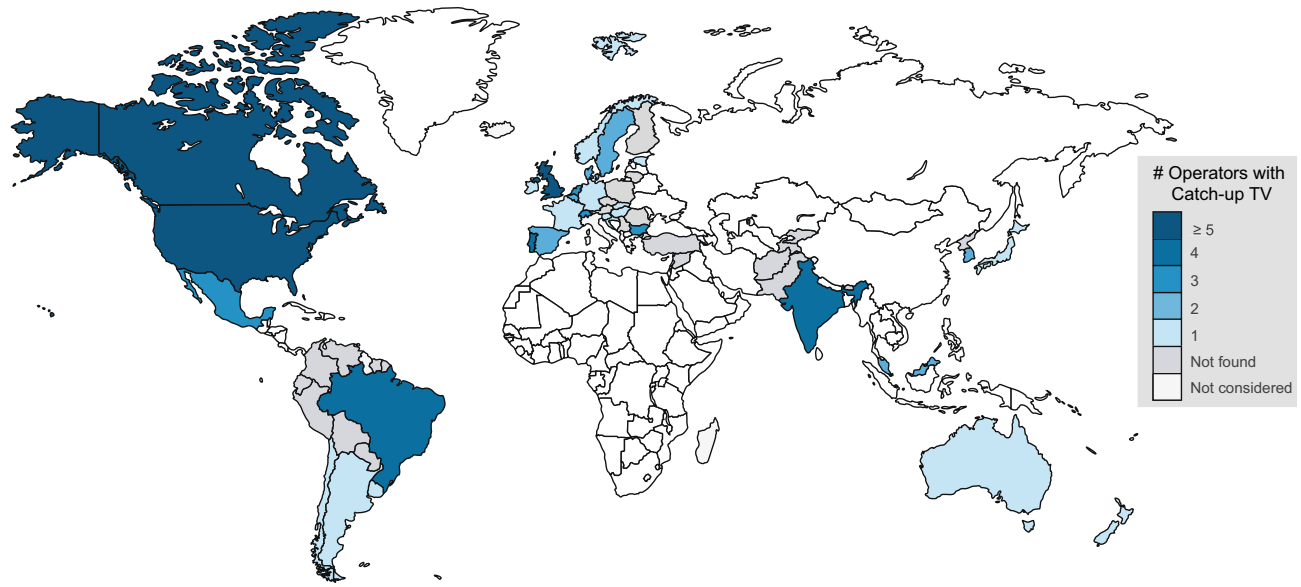


Fig. 2 Global overview of operators offering Catch-up TV and other time-shift TV services

respect to viewers' behaviors and related techno-business and social impacts of an apparent reduction of linear-TV consumption.

A thorough analysis of the worldwide overview of services offering nonlinear TV content over MONs is presented in the following sections, and in 4 separate tables: Europe (Table 2); Asia (Table 3); America (Table 4); and Oceania (Table 5).

5.1 Europe

In Europe, from the 30 countries analyzed, 20 already offer Catch-up TV services, as seen in Table 2, while 37 major Pay-TV operators offer Catch-up TV, with a prominence in England and Portugal.

With respect to countries where Catch-up TV services are not found, there are some where legal issues are an obstacle—a more detailed explanation is provided in Sect. 6.2.

The most frequent Catch-up TV time window is of 3–7 days (56 %), followed by the smallest interval of up to 3 days (28 %).

These 37 operators also support Pause-TV, while Start-over TV and DVR features are widely available. Only 6 of the considered operators offer cloud-based PVR, i.e. NPVR.

VoD services are not offered by 3 operators with Catch-up TV, which may be justified by their business models and expected users' adoption. Transaction and Subscription VoD (in most cases based on an integrated offer of Netflix) are the most common forms of VoD, whereas Electronic Sell Through VoD is only offered by 3 operators.

5.2 Asia

From the 15 Asian countries analyzed, a total of 12 operators are identified, offering Catch-up TV services in countries like India, Japan, Indonesia, Malaysia, Singapore and South Korea.

The gathered data for Asian countries is presented in Table 3, where it is possible to observe that the most frequent time window of previously aired programs is the one of 3 to 7 days (50 %). As in the American continent, a significant part (50 %) of Asian Catch-up TV services are integrated in a special section of the VoD catalog. Every operator offers T-VoD and Pause-TV features, while Start-over TV and DVR are offered by most but not all.

5.3 America

15 countries and 23 operators are analyzed in the Americas (Table 4). In most operators (83 %), Catch-up TV services comprise a selection of channels/programs integrated as a special section of the VoD catalog.

Two main differences stand out when compared to the European situation. First, the amount of available programs is smaller, as in a regular Catch-up TV service in Europe users may access most programs of the subscribed TV bundle's channels.

Furthermore, in most European Catch-up TV offers they benefit from a dedicated user interface easing the retrieval of aired programs by day of the week, channel, genre, or name.

In the Americas, the time window of previously aired programs is also dependent on the agreements with each affiliated broadcaster.

Table 2 Operators with Catch-up TV, other time-shift TV and VoD services in Europe

Country ISO 3166-3	Operator name	Catch-up TV product name	Catch-up TV window (days)					Pause-TV	Start-Over	NPVR	DVR	VoD		
			<3	[3, 7]	>7	Other	No info					T	EST	S
BEL	Proximus	TVReplay	•					•	•			•		•
BEL	Telenet	Terugkijk TV		•				•	•			•		
BGR	Vivacom	TV Archive		•				•				•		
BGR	Blizoo	Отложен старт					•	•			•			
BGR	Mtel	Тв архив	•					•	•		•			
HRV	Vipnet	Propustili ste				•		•	•	•	•	•		•
CYP	Cyta	Time Shifted TV	•					•	•			•		
DNK	Stofa	Tvarkiv	•					•	•		•	•		
DNK	Yousee	Tvarkiv					•	•	•		•	•		
GBR	TalkTalk	Catch Up TV		•				•	•		•	•		•
GBR	Youview	Catch Up TV		•				•	•		•	•		•
GBR	BT	Catch Up TV		•				•	•		•	•	•	•
GBR	Sky+	Catch Up TV			•			•	•		•	•	•	
GBR	Virgin	Catch Up TV		•				•	•		•	•		•
GBR	Starman	Ajamasin			•			•	•	•		•		•
FRA	Numericable	Replay TV de rattrapage					•	•	•	•	•	•		
DEU	Deutsche Telekom	Catch Up TV		•				•	•	•		•		•
HUN	ARCHÍV TV	Telekom TV	•					•	•			•		•
IRL	UPC	Catch Up		•				•	•		•	•		•
MLT	GO	Catch Up					•	•	•		•	•		•
NLD	KPN	Programma Gemist	•					•	•		•	•		
NLD	Caiway	Programma Gemist				•		•	•		•	•		•
NLD	Ziggo	Gemiste			•			•	•		•	•		•
NOR	Canaldigital	Ukes arkiv		•				•	•		•	•		
PRT	NOS	Gravação Automática		•				•	•	•	•	•	•	•
PRT	MEO	Gravações automáticas		•				•	•		•	•		
PRT	Cabovisão	Flashback		•				•	•		•	•		
PRT	Vodafone	Gravações automáticas		•				•	•		•	•		
SVK	SATRO	TV archív		•				•	•		•			
SVN	Amis	Na začetek	•					•	•		•	•		
ESP	Movistar	Te lo perdiste		•				•			•	•		
ESP	Telecable	Novisto/tedi		•				•	•		•	•		
SWE	CanalDigital	Veckoarkiv		•				•	•		•	•		
SWE	Com Hem	Sju dagars omstart		•				•	•		•	•		•
CHE	Swisscom	Replay		•				•	•	•		•		
CHE	Naxoo	Replay	•					•	•		•	•		
CHE	UPC	Replay	•					•	•		•	•		

Table 3 Operators with Catch-up TV, other time-shift TV and VoD services in Asia

Country ISO 3166-3	Operator name	Catch-up TV product name	Catch-up TV window (days)					Pause-TV	Start-Over	NPVR	DVR	VoD		
			<3	[3, 7]	>7	Other	No info					T	EST	S
IND	Tata Sky	Catch-Up TV		•				•			•	•		
IND	Airtel Digital TV	Timeshift		•				•	•	•	•	•		
IND	Reliance Jio	Time-Shift TV		•				•			•	•		
IND	Chitram TV	Catch-Up TV			•			•	•			•		
JPN	J:COM	On Demand			•			•	•			•		
IDN	First Media	X1		•				•	•		•	•		•
MYS	Astro	Free VoD				•		•			•	•		
MYS	HyppTV Telekom Malaysia	On-Demand Channels		•				•	•		•	•		
SGP	Singtel TV	On-Demand Channels				•		•			•	•		
SGP	StarHub Cable Vision	Catchup-TV		•				•	•		•	•		
KOR	Tbroad	Free VOD				•		•	•			•		
KOR	C&M	VOD				•		•				•		

Table 4 Operators with Catch-up TV, other time-shift TV and VoD services in America

Country ISO 3166-3	Operator name	Catch-up TV product name	Catch-up TV window (days)					Pause-TV	Start-Over	NPVR	DVR	VoD		
			<3	[3, 7]	>7	Other	No info					T	EST	S
ARG	CableVisiÓn (Grupo Clarín)	On Demand				•		•	•		•			
BRA	Net	Now				•		•	•		•	•		•
BRA	Telefônica	Multiroom VIVO TV Fibra				•		•	•		•	•		
BRA	GVT	Outra Chance GVT				•		•	•		•	•		
BRA	OI	Outra vez		•				•	•		•	•		
CAN	Rogers Communications	Rogers On Demand				•		•	•		•	•		•
CAN	Shaw	Shaw On Demand				•		•	•		•	•		
CAN	Bell	On Demand				•		•	•		•	•		
CAN	Cogeco	Cogeco On Demand				•		•	•		•	•		•
CAN	Videotron	Club Illico				•		•	•		•			•
CAN	VMedia	VCloud TV		•				•	•	•	•			•
CHL	VTR	VTR On Demand				•		•	•		•	•		
MEX	Cablevision	ONDemand				•		•	•		•	•		
MEX	Total Play	AnyTime TV				•		•	•		•	•		•
MEX	Megacable	Megacable Play				•		•	•		•	•		•
URY	Nuevosiglo	NS Now				•		•	•		•	•		
USA	Comcast	XFINITY ON DEMAND				•		•	•	•	•	•		
USA	Direct TV	72 Hour Rewind	•					•	•		•	•		
USA	Time Warner Cable	Look Back	•					•	•		•	•		•
USA	AT&T U-verse	On Demand				•		•	•		•	•		
USA	Verizon FiOS	FiOS On Demand				•		•	•	•	•	•		•
USA	Cox Communications	ON DEMAND SHOWS				•		•	•		•	•		
USA	Charter Communications	On Demand				•		•	•		•	•		•

Table 5 Operators with Catch-up TV, other time-shift TV and VoD services in Oceania

Country ISO 3166-3	Operator name	Catch-up TV product name	Catch-up TV window (days)					Pause-TV	Start-Over	NPVR	DVR	VoD		
			<3	[3, 7]	>7	Other	No info					T	EST	S
AUS	FoxTel	Anytime			•			•			•	•		
NZL	Freview	New Enhanced TV Guide			•			•			•	•		

As for the remaining time-shift TV services, all the 23 considered operators provide Pause-TV; Start-over TV and DVR. Only 3 provide a Network Personal Video Recorder.

Additionally, all but 1 operator offer VoD services, and the predominant type is T-VoD.

5.4 Oceania

In Oceania, the survey focuses on Australia and New Zealand. In these countries, two operators are found that offer a Catch-up TV service with a time window of 7 or more days (Table 5).

In addition to providing Catch-up TV services, these operators also support Pause-TV, DVR and T-VoD, although none provides Network Personal Video Recorder.

6 Discussion

6.1 Survey data analysis

This thorough overview made clear that the current worldwide footprint of the Catch-up TV service is very expressive. The technology is widely available and users' adoption shows that this is a trend with the potential to spread into other countries and operators. The survey shows that the availability time window of previously aired programs varies significantly from country to country. This effect is a direct result of the trade-off between business models, legal issues, storage, and transmission costs.

Another aspect that stands out is the fact that other time-shift TV services (Pause-TV, Start-over TV and PVR) as well as VoD services are a constant in the offers of Pay-TV operators.

The presence of all these services over MONs is proof that users value the possibility of consuming TV content at their pace in a nonlinear way, especially if they have the opportunity to easily enjoy a service like Catch-up TV, which automatically records the content they want. The large observed footprint and booming popularity of Catch-up TV services mandate a detailed analysis considering its techno-business impacts on the different stakeholders: content providers, Pay-TV providers and consumers.

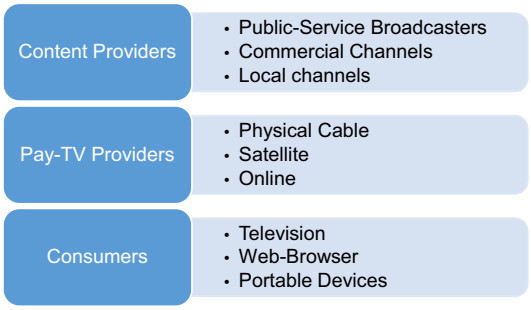


Fig. 3 Diagram of the Pay-TV industry supply chain

6.2 Techno-business impacts of catch-up TV services

Catch-up TV is the reflex of content-centric paradigms that blur the line between nonlinear and linear TV consumption. Because Pay-TV industry is supported on complex relationships between multiple stakeholders, as may be observed in Fig. 3, the decision of adding a new service must be carefully analyzed in order to consider the established balance of power, and to assess its impact along the complete supply chain, where each stakeholder is affected differently.

6.2.1 Why should catch-up TV be offered to Pay-TV customers?

The main business value proposition of Catch-up TV services lies in consumer empowerment. The control of what to watch, and when, is transferred from the broadcasters to the consumers, disrupting the established editorial control, and increasing consumer choice.

In a time where cord-cutters [43,61] are a reality, paying attention to customers is crucial to improve their satisfaction with Pay-TV services, hence fostering customer acquisition, retention, and upselling. For Pay-TV providers, preventing cord-cutting, reducing churn, and increasing the Average Revenue Per User (ARPU) is essential and requires a rich and convenient service offering. A positive impact on ARPU caused by Catch-up TV has been shown in [10].

Bjøndal [12] shows that most consumers have been clients of their Pay-TV service provider for less than 5 years, which indicates that the market is highly dynamic and that users are willing to switch providers in order to take advantage of added features, improved user experience,

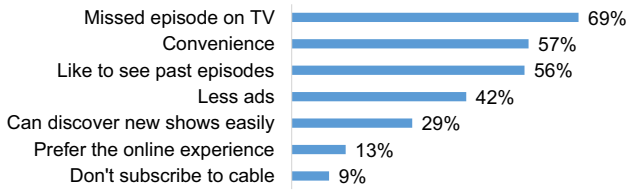


Fig. 4 Reasons to watch video online [54]

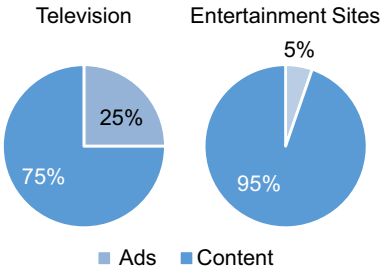


Fig. 5 Percentage of time spent watching ads [11]

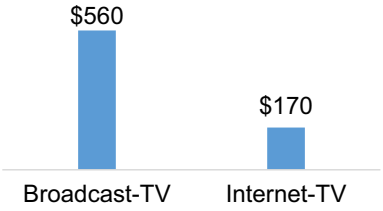


Fig. 6 The value of broadcast versus online viewers [54]. Advertising value per thousand viewers per episode

higher content quality, and lower prices. For example, Belgian operator Proximus' annualized churn rate on triple-play services was 10.5 % on its first 2015 quarter [9], up from 9.3 % on the previous quarter [8].

To determine what features present an appealing value proposition, a possible approach is to look into the reasons that drive consumers out of the Pay-TV experience into alternative media services, such as online video. ComScore data [54], displayed on Fig. 4, indicates that the main reasons for watching online content are missed TV episodes and the desire to watch past episodes of TV shows.

Broadcasters also benefit from user engagement in Pay-TV services, as the amount of advertisement watched by users and its cost is much higher than on other comparable services, such as online entertainment sites, as is clearly visible in Figs. 5 and 6.

6.2.2 Impact on Pay-TV service providers

While the benefits to consumers are well established, Catch-up TV has a significant impact on service providers' operations, presenting challenges of technical, economic, and legal nature. This is a service for the masses [17] with a high impact on the distribution infrastructures, as traditional broadcasting methods, using multicast on IPTV networks, do not work. The need to unicast video streams imposes severe network capacity requirements, which must be addressed by large investments [36]. The fact that Catch-up TV is data-intensive is also challenging, mostly because users are often not charged for the amount of data that needs to transverse the network.

In addition to technical challenges, there are also licensing issues, as content providers may impose restrictions on the content available on Catch-up TV and require additional fees. Depending on each country's legislation, and on existing agreements, adding TV channels to the Catch-up TV lineup may be challenging [53].

In countries where Pay-TV providers offer a wide range of channels in Catch-up TV, they do so on the premise that it is a kind of NPVR service, where the customer schedules full-channel recordings instead of just some shows. This seems valid for countries where NPVR shared copy is allowed and no additional compensation is due to the content owners. On the other hand, in many countries in America and Asia, where law mandates private copy [52,65], this full lineup for Catch-up TV services does not exist.

Through the survey, the authors noticed that the line-up of Catch-up TV channels offered by most American and Asian operators are limited to those with a licensing agreement with the service provider, or channels that themselves have a Catch-up TV service in place, like most free-to-air channels in France, UK or Portugal.

6.2.3 Impact on content providers

Content providers decide the content price, thus having a high bargaining power in the supply chain, which is used to leverage new delivery forms as an opportunity for increasing revenue, such as demanding micro-payments (Pay-Per-View (PPV)), or charging additional fees per delivery service. However, limiting the availability of Catch-up TV content may be counter-productive. The reasons vary depending on the TV stations' business models.

For premium TV stations, where the advertising revenue is residual and most revenue comes from user subscriptions, not allowing a service like Catch-up TV reduces its value proposition, especially if the aired content does not have any temporal relevance, which is usually the case of movies and series premium channels, but also applies to sports channels, or other TV stations where live events are particularly important.

Regarding non-premium TV stations, whose main stream of revenue originates from advertisement, the Catch-up TV proposition is also relevant. Several studies show that, in spite of a reduction in linear TV viewing, in favor of time-shifted viewing, the overall TV consumption has increased [10,49,60] due to time-shifting services. [42] shows that if

Catch-up TV were a TV station, it would be the most popular on prime-time.

Non-premium TV stations fear that the reduction on linear TV consumption will lead to a reduction on advertisement value, thus having a negative impact on revenue. However, it has been shown that not all users skip advertisements, and that advertisements get up to 44 % more views due to time-shifted viewing [47]. Additionally, the vast majority of advertisements are still relevant on most Catch-up TV reproductions, which happen mainly within 3 days of the original airing, regardless of the total Catch-up TV window [10,48].

Ultimately, because Catch-up TV increases overall media consumption, content providers get an increased exposure of their programs, and advertisements, to consumers. This motivated Nielsen [46] to release the so-called "C3" ratings that encompasses commercials watched both live and in a 3 days window, which show that some content, like serialized TV shows, get boosts of more than a full rating point. More recently, a new metric increased the commercials' analysis time window to 7 days (C7).

6.2.4 Impact on linear-TV

One of the myths regarding Catch-up TV services is that they significantly reduce the consumption of linear TV. While it has been shown that users watch less linear TV in favor of other media, the difference is not significant (−2 % over a 2 years period), and linear TV continues to be as relevant as before [49].

Even though this reduction occurs, the programs are still watched. The most popular programs in Catch-Up TV are the most watched in linear TV. Belo [10] found that prime-time content is the most watched content during prime-time and off-peak hours on nonlinear TV. This finding suggests an increased overall viewership of prime-time content in detriment of other content.

Wahlström [67] claims that Catch-up TV is a natural consequence of TV evolution. With the digitization of the production, transmission and reception, the value chain becomes flexible, allowing new features and services offerings. Thus, two consumption scenarios arise: time-shift services address content without significant temporal relevance; and linear TV focuses on programs with immediacy appeal.

TV programs may be classified into two types: long term, i.e., with lifetime spanning hours or days after broadcast; and those that require instant-audience, because their meaning and impact is lost if watched after broadcast.

The first group consists of kids shows, movies, series, soap-operas and other programs that are not usually broadcast live. The most important factor in this type of content is to occupy the idle time of the viewer, either individually, or in family.

The second group, requiring an instant-audience, relates to informative, journalistic, and live sports programs, which lose interest and relevance over time. News ceases to be news within hours, making the instantaneity a fundamental requirement of journalism.

The threshold between instant-audience and long-term programming are programs that blend entertainment with information and those with great potential to generate discussion, such as reality shows, talent shows, series season finales or the last soap-opera episode. These programs generate a better and more complete experience if watched on linear TV, by being the subject of personal conversations or social networks' shares.

One of the most important features of the TV is the ability to generate topics for discussion [23]. People often talk about what they saw on TV, whether in person or on social networks. Thus, the TV helps in audience socialization and integration.

The TV genre with the greatest potential to generate discussion subjects is sports, followed by news, and soap-operas [21]. Researchers have identified an increase of news programs that mix information with entertainment, called infotainment, and their tendency is to expand in the coming years. Hence, it is expected that linear TV will remain strong, despite the loss of audience for Catch-Up TV and other time-shift services.

6.3 Cost, performance, and technological aspects of time-shift TV services

6.3.1 Managed vs. non-managed solutions

As previously mentioned, networks used to deliver TV services may be categorized into two main classes: *managed/closed* or *unmanaged/open* [19], depending on who controls the network.

In managed networks, the delivery is performed with the involvement of providers, which ensure predetermined Quality-of-Service (QoS) features. This is the type of service that is provided on commercial IPTV platforms such as Ericsson's Mediaroom [1].

On the other hand, in an open and *uncontrolled* network the delivery takes place without any interference or quality guarantees of the providers supporting the delivery, which takes place as if it were any regular Internet content. Because the providers' networks are being used to provide a service from a third-party, which typically uses their network infrastructure for free, this type of delivery is called OTT.

The characteristics of OTT networks enable services to be delivered to the whole Internet, without any capital or operational expenditures on the network infrastructure itself,

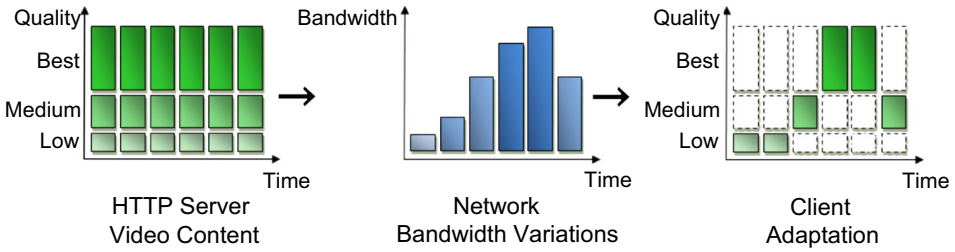


Fig. 7 HTTP Adaptive Streaming (HAS). Adapted from [2]

which are supported by the intermediate providers. There are, however, some drawbacks associated with these services: because the networks they rely on to operate are not controlled, no quality guarantees may be ensured, and the OTT providers depend entirely on the supporting best-effort network. This fact raises multiple issues as far as the users' Quality-of-Experience (QoE) is concerned [35].

The high-QoE goal requires scalable, reliable, and adaptive services, able to infer the environment conditions in *quasi*-realtime in order to provide the users with the best possible experience at a given point in time. In the context of video delivery, a good experience is correlated with metrics such as low-buffering times, no video freezes or macroblocks, a video resolution adequate to the viewing-device's screen, and in live events, low end-to-end delay, to name a few.

To address these unreliability issues associated with OTT TV delivery, TV operators have turned to HTTP Adaptive Streaming (HAS) solutions [13], depicted in Fig. 7, such as Microsoft Smooth Streaming, Apple HTTP Live Streaming (HLS), and MPEG Dynamic Adaptive Streaming over HTTP (DASH), to name a few. The idea behind these approaches is to encode the original content into streams of different quality and them fragment those streams into segments, usually 2–10 s long, that can be individually downloaded and decoded. This approach leads to very short playback start delays, and provides the added benefit of scalability and adaptability to changes in users' network conditions. MPEG DASH in particular has the added benefit of not being codec dependent and providing a clear evolutionary path from today's *HEVC* and *H.264/AAC* video/audio codecs.

6.3.2 Advantages and disadvantages of cloud time-shift TV services

From a Pay-TV's operator perspective, time-shift services stand to benefit the most from a migration from local to cloud storage, and are appealing for being cost effective, both from a capital expenditures (CAPEX) and operational expenditures (OPEX) standpoint.

The "cloudification" of STBs' functions significantly lowers the hardware costs, especially if the HDD is removed as it is responsible for a significant portion of total hardware costs, device malfunctions, and inherent maintenance.

This move also allows for a more rational use of storage resources since redundant content recorded by each client is usually only stored once. In countries where private copy law is in effect there is a reduction in the convergence gains, but the total storage requirements will still be inferior to the sum of clients' HDDs capacity in non-cloud deployments.

Migrating time-shift TV services to the cloud gives Pay-TV operators new degrees of flexibility, and does not limit evolutionary architectural changes; however, this added flexibility comes at the expense of complexity, which if not properly managed may lead to failures or errors affecting all users simultaneously.

Additionally, the dependency on always on-connectivity and remote services may significant impair the user experience in the event of network instability or bandwidth fluctuations.

7 Conclusions

This article investigates the impact of new recording technologies on the Pay-TV market. To identify the services, their differences and resources scope, a taxonomy is proposed. A global survey shows the state of the art of Catch-Up TV and other time-shift TV services in 62 countries. Finally, the article presents the techno-business impacts on the Pay-TV value chain and analyzes the impacts of Catch-up TV on linear TV.

This research brings new elements for analyzing the changing process in the TV ecosystem. For many years, people just watched linear TV, and some free-to-air channels monopolized the audience. The quality of programming was relegated to the background, since there was little choice [25].

Nowadays, with Catch-Up TV services, it is possible to choose any program recently broadcast. Thus, regardless of the transmission time, the viewer is presented with a vast choice of programs to select from. With these new technological resources, content quality becomes the differential, instead of the lack of competition or time of broadcast.

There is no longer the need for a programming schedule. Broadcasters could offer all programs simultaneously in the cloud, and the viewer would be able to choose what and when to watch, as he does with the Catch-up TV services.

Nevertheless, there is no consensus on the impact of these new recording and transmission technologies. Studies point out to a complete restructuring of TV [27,32,64]. According to these authors, it makes no sense that a handful of people running a TV station have the power to choose what millions of viewers watch. Therefore, the future of TV would be totally on-demand, online, bidirectional and programmed by the viewer. In their perspective, the end of the linear TV is a matter of time.

Other authors [15] argue that linear TV is actually gaining traction and relevance. This reasoning is based on three focal points. First, the delay in technology implementation hinders the universal access to new features for nonlinear consumption. It is possible that these services will never have universal access. Second, to use nonlinear TV services, the viewer needs to be aware of what he wants to watch and what is available. In spite of the content recommendation systems' evolution, which can solve the problem of awareness, knowing what to watch is more related to human emotions than to technology [3,24]. Third, in many cases, people do not want to be engaged in watching a TV program, they just want to have some company or background noise at home. In this case, the TV is turned on as a habit, thus the channel and program do not make a difference. Despite this uncertainty, [38] believes that most of the popular free-to-air content, especially sports and movies, will become premium content, which can be the reason for the expected Pay-TV growth in the coming years.

Although contradictory, both interpretations are relevant and possible to occur. Therefore, it is possible to conclude that the linear TV will live with the nonlinear consumption in the coming decades. There is still a demand for traditional TV, in spite of Catch-Up TV services representing a technological and market differential for Pay-TV providers. Ultimately, the viewer is rewarded with additional options to access information and entertainment.

Acknowledgments This research was supported by a Portugal 2020 grant to the UltraTV project (POCI-01-0247-FEDER-017738).

References

1. Ericsson. (2015). Ericsson mediaroom. http://www.ericsson.com/us/ourportfolio/telecom-operators/mediaroom. Accessed September, 2015.
2. Bitcodin. (2015). MPEG DASH in a Nutshell. https://www.bitcodin.com/blog/2015/04/mpeg-dash/. Accessed September, 2015.
3. Abreu, J., Almeida, P., & Teles, B. (2014). TV discovery & enjoy: A new approach to help users finding the right TV program to watch. In *Proceedings of the 2014 ACM international conference on interactive experiences for TV and online video - TVX '14* (pp. 63–70). New York: ACM Press. doi:10.1145/2602299.2602313. http://dl.acm.org/citation.cfm?doid=2602299.2602313.
4. Abreu, J., Becker, V., & Nogueira, J. (2015). Overview of Catch-up TV and other time-shift TV services. http://socialitv.web.ua.pt/wp-content/uploads/2016/03/Non_Linear_TV_Worldwide.pdf. Accessed September, 2015.
5. Alcatel-Lucent Cloud DVR. (2015). https://www.alcatel-lucent.com/solutions/cloud-dvr. Accessed September 2015.
6. Asquith, K., & Hearn, A. (2012). Promotional prime time: Advertainment, internal network promotion, and the future of canadian television. *Canadian Journal of Communication37*, 241–257. http://www.cjc-online.ca/index.php/journal/article/view/2494.
7. Beauvisage, T., & Beuscart, J.S. (2012). Audience dynamics of online catch up TV. In *Proceedings of the 21st international conference companion on World Wide Web - WWW '12 Companion* (p. 461). New York: ACM Press. doi:10.1145/2187980.2188077. http://dl.acm.org/citation.cfm?doid=2187980.2188077.
8. Belgacom: Proximus Q4 Quarterly Report. (2014). http://www.proximus.com/sites/default/files/Documents/Investors/Reports/2014/en/q4/Belgacom_Q4_2014.pdf. Accessed Sepember, 2015.
9. Belgacom: Proximus Q1 Quarterly Report. (2015). http://www.proximus.com/sites/default/files/Documents/Investors/Reports/2015/en/Q12015_rapport.pdf. Accessed September, 2015.
10. Belo, R., Godinho de Matos, M., & Ferreira, P. (2013). Prime-time any time: The effect of time-shifted tv on media consumption. *SSRN Electronic Journal*. doi:10.2139/ssrn.2242531.
11. Berman, S.J., Battino, B., & Feldman, K. (2010). Beyond content: Capitalizing on the new revenue opportunities. http://public.dhe.ibm.com/common/ssi/ecm/gb/en/gbe03361usen/GBE03361USEN.PDF. Accessed September, 2015.
12. Bjøndal, T.S., & Gedde, M. (2011). Ubiquitous TV: A business model perspective on the Norwegian Television Industry. Master, Norwegian University of Science and Technology. http://brage.bibsys.no/xmlui/handle/11250/266027.
13. Broadpeak: Video Recording in the Cloud: Use Cases and Implementation. (2015). http://www.broadpeak.tv/upload/produit/fichier/35-605-broadpeak_cloudpvr_whitepaper_2015_09.pdf. Accessed January, 2016.
14. Bury, R., & LI, J. (2015). Is it live or is it timeshifted, streamed or downloaded? Watching television in the era of multiple screens. *New Media & Society*, *17*(4), 592–610. doi:10.1177/1461444813508368.
15. Cannito, N. (2010). *A televisão na era digital: Interatividade, convergência e novos modelos de negócio*. São Paulo: SUMMUS.
16. Chaney, A.J., Gartrell, M., Hofman, J.M., Guiver, J., Koenigstein, N., Kohli, P., & Paquet, U. (2014). A large-scale exploration of group viewing patterns. In *Proceedings of the 2014 ACM international conference on Interactive experiences for TV and online video - TVX '14*. New York: ACM Press. doi:10.1145/2602299.
17. CNC: L ' économie de la télévision de rattrapage en 2014. Centre national du cinéma et de l'image animée pp. 1–33 (2015). http://www.cnc.fr/web/fr/ressources/-/ressources/6592632.
18. Comcast: XFINITY TV Go. (2015). http://tvgo.xfinity.com/apps.
19. Cooper, W., & Lovelace, G. (2007). Delivering audio and video over broadband. In *IPTV conference 2007 - deployment and service delivery*, IET, pp. 1–66.
20. De Vleeschauwer, D., Avramova, Z., Wittevrongel, S., & Bruneel, H. (2009). Transport capacity for a catch-up television service. In *Proceedings of the seventh european conference on European interactive television conference - EuroITV '09* (p. 161). New York, NY: ACM Press. doi:10.1145/1542084.1542117. http://portal.acm.org/citation.cfm?doid=1542084.1542117.
21. Dezfuli, N., Khalilbeigi, M., Mühlhäuser, M., & Geerts, D. (2011). A study on interpersonal relationships for social interac-

tive television. In *Proceedings of the 9th international interactive conference on Interactive television - EuroITV '11* (p. 21). New York, NY: ACM Press. doi:10.1145/2000119.2000123. http://portal.acm.org/citation.cfm?doid=2000119.2000123.
22. Elemental: Fast Forward: Implementing Live-to-VOD Services. (2015). http://www.digitaltveurope.net/309572/fast-forward-implementing-live-to-vod-services/. Accessed September, 2015.
23. Gauntlett, D. (1999). TV living: Television, culture and everyday life. Routledge. https://books.google.pt/books?id=dWMdhtyFAcEC.
24. Gorton, K. (2009). Media audiences: Television, meaning and emotion. Edinburgh University Press Series. Edinburgh University Press. https://books.google.pt/books?id=UWEMD13YmIMC.
25. Gruenwedel, E. (2015). Comcast Controls 15% of EST Market, Other MVPDs to Enter. http://www.homemediamagazine.com/digital-evolution/lionsgate-ceo-comcast-controls-15-est-market-other-mvpds-enter-32521. Accessed September, 2015.
26. Hess, J., Ley, B., Ogonowski, C., Reichling, T., Wan, L., & Wulf, V. (2012). New technology@home. In *Proceedings of the 10th European conference on interactive tv and video - EuroiTV '12* (p. 185). New York: ACM Press. doi:10.1145/2325616.2325653. http://dl.acm.org/citation.cfm?doid=2325616.2325653.
27. Jaffe, J. (2008). *O declínio da mídia de massa. Por que os comerciais de TV de 30 segundos estão com os dias contados*. São Paulo: MBooks.
28. Jenkins, H. (2006). Convergence culture: Where old and new media collide. New York: NYU Press. https://books.google.pt/books?id=RlRVNikT06YC.
29. Jennes, I., Pierson, J., & Van den Broeck, W. (2014). User empowerment and audience commodification in a commercial television context. *The Journal of Media Innovations*, *1*(1), 71–87. doi:10.5617/jmi.v1i1.723.
30. Clover, J. (2014). Counting Netflix by country. http://www.broadbandtvnews.com/2014/07/24/counting-netflix-by-country/. Accessed September, 2015.
31. Juniper Research: OTT - A Threat Networks Can't Shake Off. (2015). http://www.juniperresearch.com/document-library/white-papers/ott-a-threat-networks-cant-shake-off. Accessed September, 2015.
32. Kackman, M., Binfield, M., Payne, M.T., Perlman, A., & Sebok, B. (2010). Flow TV: Television in the Age of Media Convergence. Routledge, New York. https://books.google.pt/books?id=unaQAgAAQBAJ.
33. Kalia, S. (2014). DVR and its impact on indian market: now and in future. SAGE Open. doi:10.1177/2158244014560551.
34. Lange, A., Benhamou, N., Joux, A., Gros, H., & Guen, J.M.L. (2009). Video on demand and catch-up tv in europe. http://www.obs.coe.int/documents/205595/264625/VOD+2009+EN.pdf. Accessed September, 2015.
35. Larbey, P., Mestric, R., Robinson, D., & Thomas, C.: The future of IP video: From Pay TV to Cloud TV (2014). http://resources.alcatel-lucent.com/asset/176115. Accessed January, 2016.
36. Li, Z., & Simon, G. (2011). Time-shifted TV in content centric networks: The case for cooperative in-network caching. In *2011 IEEE international conference on communications (ICC)* (pp. 1–6). IEEE. doi:10.1109/icc.2011.5963380. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5963380.
37. Mann, F., Mahnke, R., & Hess, T. (2012). Find your niches: A guide for managing intermedia effects among content distribution channels. *International Journal on Media Management*, *14*(4), 251–278. doi:10.1080/14241277.2012.706763.
38. Medina, M., Herrero, M., & Etayo, C. (2015). The impact of digitalization on the strategies of pay TV in Spain. Tech. rep., Sociedad Latina de Comunicación Social, La Laguna, Tenerife. doi:10.4185/RLCS-2015-1045en. http://www.revistalatinacs.org/070/paper/1045na/15en.html.
39. MEO: MEO GO. (2015) http://meogo.meo.pt/.
40. Minoli, D. (2012). Linear and non-linear video and TV applications: Using IPv6 and IPv6 Multicast. Hoboken: Wiley. Accessed September, 2015.
41. Mohan, D. (2014). *The evolving value chain in the television industry : Changes in pay TV delivery and its implications for the future*. Ph.D. thesis, Massachusetts Institute of Technology. http://hdl.handle.net/1721.1/90718.
42. Moulding, J. (2014). Swisscom explains push for NPVR, as it heads towards 50 % time-shift viewing. http://www.v-net.tv/swisscom-explains-push-for-npvr-as-it-heads-towards-50-time-shift-viewing. Accessed September, 2015.
43. Murray, S. (2014). OTT to reach nearly half the world's TV households by 2020. Tech. rep., Digital TV Research. https://www.digitaltvresearch.com/ugc/OTTHH2014TOC_toc_105.pdf. Accessed September, 2015.
44. Nencioni, G., Sastry, N., Chandaria, J., Crowcroft, J., Nishanth, S., Chandaria, J., & Crowcroft, J. (2013). Understanding and decreasing the network footprint of Catch-up TV. In *Proceedings of the 22nd international conference on world wide web* (p. 12). Rio de Janeiro: International World Wide Web Conferences Steering Committee. http://dl.acm.org/citation.cfm?id=2488388.2488472.
45. Netflix: Where is Netflix available. (2015). https://help.netflix.com/en/node/14164. Accessed September, 2015.
46. Nielsen. (2009). C3 TV ratings show impact of DVR ad viewing. http://www.nielsen.com/us/en/insights/news/2009/c3-tv-ratings-show-impact-of-dvr-ad-viewing.html. Accessed September, 2015.
47. Nielsen. (2010). State of the media: DVR use in the U.S. http://www.nielsen.com/content/dam/corporate/us/en/newswire/uploads/2010/12/DVR-State-of-the-Media-Report.pdf. Accessed September, 2015.
48. Nielsen. (2011). Time Shift Viewing - Setting the scene for 2012. http://www.thinktv.co.nz/wp-content/uploads/TSV-Charts2_Part11.pdf. Accessed September, 2015.
49. Nielsen. (2014). The digital consumer. http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2014Reports/the-digital-consumer-report-feb-2014.pdf. Accessed September, 2015.
50. Noam, E. (2014). Cloud TV: Toward the next generation of network policy debates. *Telecommunications Policy*, *38*(8–9), 684–692. doi:10.1016/j.telpol.2013.10.004.
51. Olswang. (2013). Content meets the cloud: What is the legality of cloud TV. http://www.olswang.com/media/34009878/go_230_cloudpvr_v4_lo-res.pdf. Accessed September, 2015.
52. Olswang. (2014). Content meets the cloud: Aereo and the future of cloud TV. http://www.olswang.com/media/48210818/aereo_report.pdf. Accessed September, 2015.
53. Picard, R.G., Davis, C.H., Papandrea, F., & Park, S. (2015). Platform proliferation and its implications for domestic content policies. *Telematics and Informatics*. doi:10.1016/j.tele.2015.06.018.
54. Piech, D. (2010). The state of online video. http://www.comscore.com/content/download/7235/125253/version/1/file/comScore+OMMA+Video+Presentation+-+Jan+2011.pdf. Accessed September, 2015.
55. Poggi, J. (2015). Apple, Sony, Dish Push OTT viewing over the top. http://adage.com/article/media/apple-sony-dish-push-ott-viewing-top/297712/. Accessed September, 2015.
56. Saxbe, D., Graesch, A., & Alvik, M. (2011). Television as a social or solo activity: Understanding families' everyday television viewing patterns. *Communication Research Reports*, *28*(2), 180–189. doi:10.1080/08824096.2011.566104.
57. Simplestream Ltd: TVPlayer. (2015). https://tvplayer.com/.
58. Sling television: Sling TV. (2015). https://www.sling.com/.

59. Sony: PlayStation Vue. (2015). https://www.playstationnetwork.com/vue/.
60. ThinkTV: PVRs drive incremental audiences. (2015). http://www.thinktv.com.au/Media/Stats_&_Graphs/2014/PVRs_drive_incremental_audiences.pdf. Accessed September, 2015.
61. Tice, D. (2014). Adventures in cord-cutting. http://blog.gfk.com/2014/10/adventures-in-cord-cutting/. Accessed September, 2015.
62. Time Warner Cable: TWC TV. (2015). http://www.timewarnercable.com/en/tv/features/twc-tv.html.
63. Tsekleves, E., Whitham, R., Kondo, K., & Hill, A. (2011). Investigating media use and the television user experience in the home. *Entertainment Computing*, *2*(3), 151–161. doi:10.1016/j.entcom.2011.02.002.
64. Turner, G., & Tay, J. (2009). *Television studies after TV: Understanding television in the post-broadcast era* (1st ed.). New York: Routledge.
65. United States Court of Appeals for the Second Circuit: The Cartoon Network LP, LLLP v. CSC Holdings, Inc. pp. 1–44 (2007). http://www.ca2.uscourts.gov/decisions/isysquery/339edb6b-4e83-47b5-8caa-4864e5504e8f/1/doc/07-1480-cv_opn.pdf.
66. Vanattenhoven, J., & Geerts, D. (2015). Broadcast, video-on-demand, and other ways to watch television content. In *Proceedings of the ACM international conference on interactive experiences for TV and online video - TVX '15* (pp. 73–82). New York, NY: ACM Press. doi:10.1145/2745197.2745208. http://dl.acm.org/citation.cfm?doid=2745197.2745208.
67. Wahlström, M. A., & Kankainen, A. (2011). Digital TV transition and the hard disk drive revolution in television viewing Helsinki Institute for Information Technology HIIT. *International Journal of Communication*, *5*, 1606–1622.
68. Walt Disney Movies: Video On Demand & Pay Per View. (2015). http://disney.go.com/vodppv/vod.html. Accessed September, 2015.
69. Wikipedia: List of cable television companies. (2015). http://en.wikipedia.org/wiki/List_of_cable_television_companies. Accessed September. 2015.
70. Williams, R., & Williams, E. (2003). Television: Technology and cultural form. Routledge classics. London : Routledge. https://books.google.pt/books?id=9XYfPRBR3awC.
71. Wirtz, B. W. (2014). Business models, value chains and competencies in media markets. A service system perspective. *Palabra Clave - Revista de Comunicación*, *17*(4), 1041–1066. doi:10.5294/pacla.2014.17.4.3.

Jorge Abreu got his graduation and Master's degree in Electronic and Telecommunication by the University of Aveiro, Portugal. After his participation in several European projects he joined the Department of Communication and Arts and concluded is PhD in Sciences and Communication Technologies. Currently, he teaches in the undergraduate course of New Communication Technologies, in the Master in Multimedia Communication (where is member of its Scientific Committee) and in the PhD program in Information and Communication on Digital platforms. As a member of the research unit DigiMedia (Digital Media and Interaction), he has been the scientific coordinator and PI of a wide range of international and national research projects in the interactive TV area, new media, and cross-platform content, in partnership with several entities including: the Foundation for Science and Technology, Ministry of Science and Technology, Commission of the European Union and Portugal Telecom. Some of those projects have been developed by his research team in Social iTV (http://socialitv.web.ua.pt/).

João Nogueira is a PhD student at University of Aveiro and R&D engineer at Portugal Telecom Inovação, SA since 2010. He received his Master's in Electronics and Telecommunications from University of Aveiro in 2010, with relevant contributions to the FP7 European project 4WARD. He joined Portugal Telecom Inovação, SA – now Altice Labs, SA - and focused his work on cloud multimedia services and interactive applications for IPTV. He is actively involved the "Next generation Over-The-Top multimedia Services" (NOTTS) Eureka Celtic Plus Project since 2013, and is responsible for key consortium contributions. His current research goals include the improvement of end-users' quality of experience in the context of multimedia Over-The-Top delivery, including Live TV, Video On-Demand, and Catchup-TV services.

Valdecir Becker is a journalist, Master of Engineering and Knowledge Management (2006 UFSC) and Doctor of Science (Electrical Engineering, 2011, USP). He is a Professor at the Science Computer Center and Postgraduate Program in Journalism at the Federal University of Paraíba (UFPB); researcher at LAViD (Digital Video and Applications Center); and member of the editorial board of the Brazilian Society of Television Engineering (SET). Over the years, he has participated in several academic projects related to Digital TV, studying new formats of convergent and multi-platform content and the impact of digital technologies in content and business models. He has written books and papers about digital TV, interactivity, HCI, audience and reception studies.

Bernardo Cardoso received his Master of Information Management degree at Universidade de Aveiro in 2002, and his Bachelors in Accounting Auditing also in Aveiro, at the Instituto Superior de Contabilidade e Administração, in 1999. He joined Portugal Telecom Inovação, SA (now Altice Labs, SA) in 2000 and was integrated in the Multimedia Technologies group, where he participated in several digital television, video, and interactive projects, with a particular focus on MPEG and DVB systems. He had a prominent role in the interactive television project of TV Cabo, regarding interactive applications, contents, testing, and performance analysis. Recently, he has been involved in IPTV related projects, having contributed to the customization and application development of the Meo IPTV platform. He has also had a relevant participation in both National and European projects in the interactive TV area, such as: IST-2001-34861 GMF4iTV, IST-4-027098 porTiVity, IST-2010-248652 ALICANTE, and the SI IDT Smart@Home. In his current role, he leads the Interactive and Digital, Internet and Television department, responsible for projects in several areas, including Interactive Television, Mobile and OTT Television, Added Value Services, Healthcare, eLearning, Online Payments, Personal Cloud Storage, User Experience and Usability.

Appendix C

Catch-up TV Analytics: Statistical Characterization and Consumption Patterns Identification on a Production Service.

Catch-up TV analytics: statistical characterization and consumption patterns identification on a production service

João Nogueira, Lucas Guardalben, Bernardo Cardoso & Susana Sargento

Multimedia Systems

ISSN 0942-4962

Multimedia Systems
DOI 10.1007/s00530-016-0516-7

Springer

Multimedia Systems
DOI 10.1007/s00530-016-0516-7

REGULAR PAPER

Catch-up TV analytics: statistical characterization and consumption patterns identification on a production service

João Nogueira[1,2] · Lucas Guardalben[3,4] · Bernardo Cardoso[1] · Susana Sargento[3,4]

Received: 14 January 2016 / Accepted: 19 April 2016

Abstract Multimedia IP Television services, such as on-demand Catch-up TV, are in an active migration process towards Over-The-Top (OTT) delivery using state-of-the-art Content Delivery Networks (CDNs). Maintaining the same Quality-of-Experience (QoE) of managed IPTV networks is challenging and requires a thorough understanding of users' behaviors and content demand characteristics. This article leverages Catch-up TV usage logs obtained from a Pay-TV operator's live production IPTV service containing over 1 million subscribers to characterize and extract insights from service utilization at a scale and scope not yet addressed in the literature. A detailed analysis on the characteristics of users' viewings is performed, with a study of when, where, and how often users access the service, along with how they behave during each viewing session. The results show that Catch-up TV consumption exhibits very high levels of utilization throughout the day, and is heavily polarized towards specific genres, recently aired programs, and content broadcasted during prime-time. The *superstar* effect is notorious. This analysis is complemented by a service optimization perspective, which shows that large gains are achievable by caching popular programs, and by loading content in advance to users' Set-Top-Boxes (STBs). This comprehensive research study is supplemented by detailed statistical information tables, which highlight the feasibility of efficiently migrating Catch-up TV services to OTT-scenarios, and provide the foundations for future works able to explore these results.

Keywords Catch-up TV · IPTV · OTT Multimedia · Production data · Statistics

1 Introduction

Large-scale delivery of Catch-up TV content represents one of the biggest challenges of Pay-TV operators [4, 11], mostly due to two reasons: first, the content must be streamed in unicast to each client, with dedicated connections per user; second, Catch-up TV content demand is several orders of magnitude larger than that of traditional Video-on-Demand (VoD) content [9].

The fact that Catch-up TV is data intensive is challenging as it is usually provided as a supplement to Pay-TV subscriptions with no added cost. The network impact of Catch-up TV is expected to keep growing with its popularity, which has been one of the main drivers of an increase in the average time spent by users watching TV [16]. To keep up with a growing demand, IPTV operators are turning to OTT delivery solutions which do not require investments on managed IPTV infrastructure, and increase the reach of services that may have been previously limited to certain geographic areas.

However, this move requires overcoming several challenges. Given the different requirements of OTT delivery, when compared to that of managed networks, a thorough

Communicated by A. U. Mauthe.

This research was funded by UltraTV (Portugal 2020 POCI-01-0247-FEDER-017738) and GAPOTT projects (IAPMEI, QREN and COMPETE 2013/34009).

✉ João Nogueira
joaonogueira@ua.pt

[1] Altice Labs, SA, Aveiro, Portugal

[2] University of Aveiro, Aveiro, Portugal

[3] Instituto de Telecomunicações, Aveiro, Portugal

[4] University of Aveiro, Aveiro, Portugal

Published online: 05 May 2016

understanding of service usage is required to properly decide on OTT CDN architectures, plan the physical and logical location of clusters and replica servers, tune caching algorithms, select optimal request routing mechanisms, and estimate computational, network and storage requirements, to name a few.

In addition to OTT-specific service improvements derived from utilization data, the characterization of Catch-up TV consumption presents several optimization opportunities, both from users' and operators' standpoints, regardless of the delivery approach. An optimized service improves users' QoE and overall service satisfaction, which is essential to prevent churn in modern and highly competitive Pay-TV markets.

From an operator's viewpoint, a thorough understanding of content consumption patterns fosters savings on both Capital Expenditures (CAPEX) and Operational Expenditures (OPEX). CAPEX may be reduced by investing on less extra capacity, because the exact service requirements are known and the delivery system is optimized to meet them, which also contributes to reducing the OPEX.

Other potential savings come in the form of energy efficiency, achievable using elastic resources taking advantage of content consumption patterns to provision only the required resources, or even by loading content in advance into users devices with the purpose of lowering peak resource demand. In summary, an exhaustive modeling of Catch-up TV content consumption patterns enables a great deal of optimization opportunities, and is thus the focus of this work. This characterization is supported on a Catch-up TV consumption dataset acquired from a production service.

The remainder of this study starts by examining the related work on Sect. 2 and proceeds to describing the available data set in Sect. 3. Next, in Sect. 4, a detailed characterization and discussing of Catchup-TV services is performed. The paper is wrapped up in Sect. 5, where the main conclusions are presented, followed by the future work.

Appendix A provides tables with detailed statistical information of the analyses conducted in this study.

2 Related work

The importance of valued-added services in the Pay-TV ecosystem has been established by several industrial and scientific works, which explore their impact in linear/live TV viewing and on advertisements' viewings as well [9, 13, 14, 16].

Among nonlinear IPTV services, Catch-up TV distinguishes itself as the most popular one, even surpassing the popularity of "classical" VoD services such as Transaction VoDs (T-VoDs) or Electronic Sell Through VoD (EST-VoD) [5, 9]. As a consequence of its massive popularity, Catch-up TV imposes a severe strain on the delivery infrastructure, and has motivated several authors to tackle modeling and optimization challenges.

The work in [6] provides behavioral insights on online Catch-up TV audience, derived from a dataset of French TV consumption. The paper's conclusions contradict the *long-tail* effect hypothesis, and show that most Catch-up TV consumption refers to recently broadcast programs, hence suggesting that this service blurs the frontier between linear and nonlinear TV consumption, and does not break the synchrony of live TV. In spite of these pertinent conclusions, the study is limited to 11,682 videos available on a 5-month window and to 7 TV channels. Additionally, given the restricted service availability on online platforms, the results may not be directly applicable to the IPTV scenario, which provides an integrated TV experience, and facilitates switches from linear to nonlinear TV.

The symbiosis between Catch-up TV and other TV services with peoples' habits is explored in [18], where a survey is conducted to understand *when*, *how*, and *why* users resort to these services, and how they fit together with their daily routines.

In addition to the works focusing on service characterization and modeling, other research bodies concentrate on optimization challenges from a content caching perspective, and its impacts on the bandwidth requirements from the origin servers.

In [15], the authors take advantage of a large dataset from a popular online Catch-up TV service to explore optimization opportunities by prefetching content into the clients' devices to reduce peak bandwidth consumption. In doing so, several conclusions are withdrawn regarding how users behave. In addition to showing a high engagement, users access the service in time-spread manner throughout the day, and exhibit strong preferences for a small set of programs.

A complementary work is performed in [1], where Abrahamsson et al. provide an empirical IPTV work model based on a realistic scenario simulation which considers the large discrepancies in popularity, with the purpose of evaluating the performance of traditional caching algorithms, including Least Recently Used (LRU) and Least Frequently Used (LFU), and estimating the bandwidth requirements of time-shift services. The study's conclusions demonstrate that LFU is the most favorable caching approach; however, the study neglects the fact that Catch-up TV content has a life-time expectancy that must be taken into account, so that popular content that is no longer valid does not prevent new content from populating the caches.

Another study is conducted in [3] regarding a *TV-on-Demand* service providing Catch-up TV, T-VoD, and Subscription VoD (S-VoD) content. In spite of the mixed service-type analysis, this study's conclusions support the occurrence of the Pareto-principle, or the 80–20 rule, whereby the 20 % most popular assets are responsible for 80 % of the total content requests. Research is also conducted on the content *cacheability*, which is shown to be very high even when using traditional caching algorithms such as LRU and LFU.

This work is improved in [2], where additional effects are exploited, such as program popularity variability with time. A characterization of its decay with time and genre is also provided. The results show that the content genre and the Catch-up TV availability window plays a very important role on the performance of caching algorithms and, therefore, on the streaming bandwidth required from the origin servers.

In addition to the research works focused on Catch-up TV, other measurement studies exist that characterize and model key aspects of IPTV services such as linear/live TV, and T-VoD services. In the work by Cha et al. [7], the users' live TV channel changing behavior is exhaustively analyzed. The work's chief conclusions indicate that most channel switching events happen within 10 s, suggesting that users' have a very volatile focus. Other key findings pertain to the channels' popularity, which is found to change with the time of day, and to daily viewing patterns, which vary with the channels' genres. Gopalakrishnan et al. [12] leverage traces across a 2-year period from a large-scale IPTV service to provide models for the video request *arrival process* and *stream control* of a T-VoD service. A detailed characterization shows that VoD assets may be grouped into five separate clusters of video lengths that the video popularity distribution follows an approximate Zipf distribution, and that a strong popularity drop-off exists as the content ages, showing that a content's recency influences its popularity.

As a whole, to the best of the authors' knowledge, the existing literature on Catch-up TV consumption studies relies on relatively small datasets and is mostly focused on online Catch-up TV consumption, even though a large portion of this service's usage happens on Pay-TV integrated scenarios. Moreover, these studies provide fragmented analyses on different aspects of Catch-up TV consumption and do not provide reference statistical data for posterior use by the scientific community, hence the need for a thorough and large-scale Catch-up TV characterization.

3 Dataset description

A Catch-up TV consumption dataset is collected from a major IPTV operator and contains 30 days of program request logs, regarding the full month of April 2015.

This nonlinear service provides free access to the previous 7 days of program airings on 80 TV channels, depending on users's subscriptions. The content is delivered through a managed network infrastructure using RTSP streams. Even though it would be desirable to have information on users' genre and age, the fact that the TV is commonly shared by several family members, and that the IPTV service in question does not support user profiles, prevents a targeted analysis.

Each request log entry enables a rich characterization of an individual playback session. Any information that might reveal user details is anonymized. Time and date fields are in Greenwich Mean Time (GMT) timezone. Each item has the following form:

- *Account Id*—Unique user account identification, which enables household-level behavior tracking;
- *Set-Top-Box (STB) Id*—Unique STB identification to distinguish requests from different devices in the same household;
- *District*—Geographical information containing the household location (district);
- *Title*—Name of the requested program;
- *Station Id*—Unique identification of the TV station which aired the requested program;
- *Station Genre*—Classification of the TV station main genre. Falls under the following categories: *General*, *Sports*, *Kids*, *Documentaries*, *News*, *Movies And Series*, and *Entertainment*;
- *Station Video Quality*—Video quality indication of the TV channel: either High Definition (HD) or Standard Definition (SD);
- *Program Id*—Unique identification of the program within the Electronic Programming Guide (EPG);
- *Series Id*—Unique identification of a TV series within the Electronic Programming Guide (EPG);
- *Season Number*—If the program is a TV Series, its season number;
- *Episode Number*—If the program is a TV Series, its episode number;
- *Start Time*—Original broadcasting start time of the requested program, as per the EPG;
- *End Time*—Original broadcasting end time of the requested program, as per the EPG;
- *Play Time*—Timestamp of a playback session start, i.e., when the user requested the program.

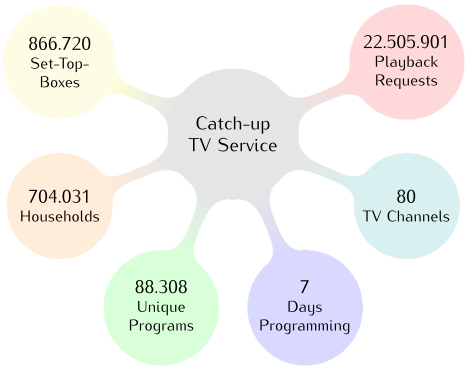


Fig. 1 Catch-up TV dataset: key data indicators

– These data fields are sufficient to extrapolate additional information, such as the playback day of week and content duration, for example.

3.1 Data cleaning

Considering that the raw data are generated from systems that may be unreliable, produce duplicate entries, and contain records from test accounts, an initial data cleaning process is performed to remove data that do not accurately reflect the service usage:

– Removal of data originated from test accounts;
– Removal of duplicate entries;
– Dates and times are adjusted to the Portuguese mainland time zone.
– After performing these data cleaning procedures, the key data indicators for the available dataset were extracted and are presented in Fig. 1.

4 Dataset analysis

The analysis performed in this section aims to provide meaningful insights onto content demand patterns from Catch-up TV service logs. This study is subdivided into four key topics: *Program Corpus vs. Program Requests*, *Service Utilization*, *Viewing Sessions Characterization*, and *Content Delivery Optimization*.

The first topic starts with a high-level characterization of the programs available for consumption, i.e., the program corpus, and proceeds to performing comparisons with what is actually requested by users. Next, a service utilization analysis takes a look at *when*, *where*, and *how* users utilize the service, according to their geographical location, number of client devices, and hour of day, to name a few. Continuing with the *Viewing Sessions Characterization* section, a more detailed examination is performed from the perspective of each individual viewing session, to study how users behave, how much time they spend on each interaction, and how long they take to settle on a particular content. Lastly, section *Content Delivery Optimization* seeks to determine if there are content demand characteristics that may be taken advantage of by caching or prefetching algorithms, with the purpose of improving the service, both from the operator's and the clients' standpoint.

To clarify the terms used throughout this study, the following concepts are defined:

– *Program Corpus*—The complete set of available Catch-up TV programs at any given time;
– *Program Request*—User action that initiates a Catch-up TV program viewing;
– *Viewing Session*—A set of back-to-back program requests. A viewing session contains at least one program request;
– *Original Airing Time*—Time at which a Catch-up TV program was broadcasted on linear TV, i.e., a timestamp such as 20:00:00 01-04-2015;
– *Request Time*—Time at which a user performed a Program Request, i.e., a timestamp;
– *Weekdays*—Monday through Friday;
– *Weekend*—Saturday and Sunday.

Given the large amounts of information presented in this analysis, a choice was made to provide the summary statistical information of the ensuing characterizations in tables grouped together on Appendix A.

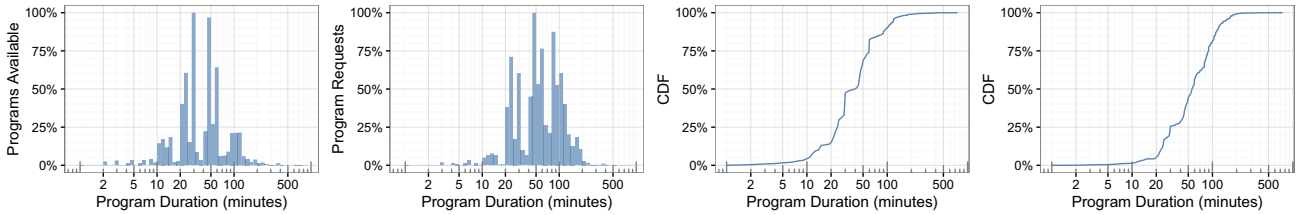
When pertinent, the data presented in the figures are normalized so that 100 % represents the maximum value, and 0 % the minimum value. This normalization maintains the proportionality relationship between the multiple values and does not affect a critical analysis, but avoids disclosing absolute numbers.

4.1 Program corpus vs. program requests

To understand the differences between the complete set of available content—*the program corpus*—and the *program requests*, this section provides a set of key comparisons that sheds light into how well adjusted the content offer is with user demand.

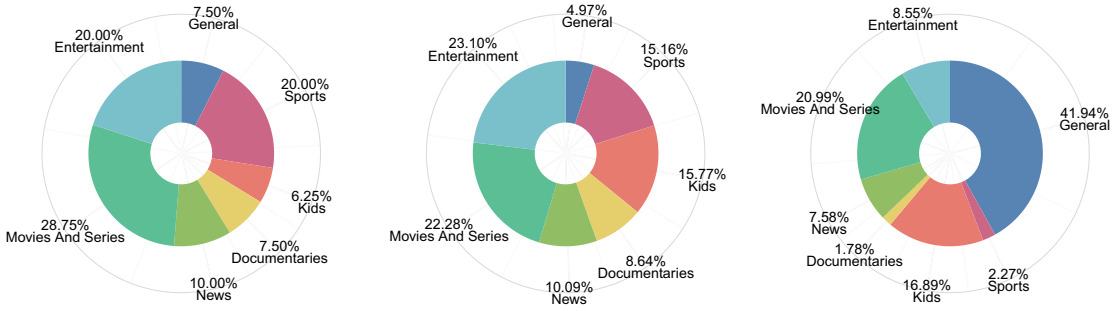
4.1.1 Content duration distribution

This evaluation focuses on a content duration perspective. The plots in Fig. 2 provide a dual perspective on the programs' durations, in the form of histograms and CDFs, and use a logarithmic *x*-axis scale, which corresponds to the programs' durations for clarity reasons. The statistical summary data for this analysis are presented in Table 1.



(a) Corpus Duration Hist. **(b)** Requests Duration Hist. **(c)** Corpus Duration CDF. **(d)** Requests Duration CDF.

Fig. 2 Program duration characterization: corpus vs. requests



(a) TV Channels. **(b)** Program Corpus. **(c)** Program Requests.

Fig. 3 Content genre distribution

Starting with Fig. 2a, b, the first main observation is the disparity and range of program durations. Four main lobes are clearly identifiable and comprise programs with approximate durations of 10 min, 20–30 min, 40–60 min and 80–120 min, which is to be expected when considering typical EPGs and program runtimes (kids, small series, long series and movies, respectively). A few long programs, with over 120 min, exist but are not very common.

Figure 2c, d provides a complementary viewpoint on the previous conclusions, by showing that the steepest plot curves refer to programs with a duration from 10 min up to approximately 100 min.

The comparison between program requests and corpus shows that users favor longer programs. In the case of Fig. 2a, b, this effect is shown by the higher mass of programs in the upper duration ranges, while the CDF of Fig. 2d is shifted to the right when compared to Fig. 2c, towards longer duration content.

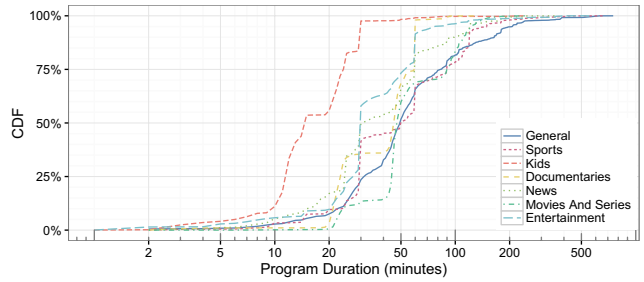
4.1.2 Content genre distribution

By leveraging existing channel genres classification, a complementary angle on the available and requested programs is provided. The available metadata does not individually classify each individual program, hence the broader channel genre is used. While this classification may not be accurate given that, for example, generalist TV stations broadcast content of multiple genres, the classification data available do not enable a more fine-grained, per-program, analysis.

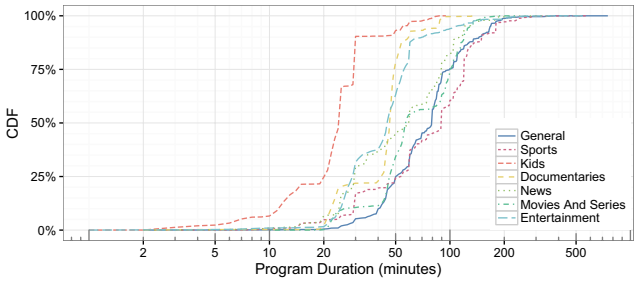
Figure 3 uses pie-charts to present three different overviews on the distribution of available and requested programs by channel and content genre. The distribution of channels by genre is illustrated in Fig. 3a, while Fig. 3b focuses on available programs with respect to their TV channel main genre. Finally, Fig. 3c presents the genres distribution of the actually requested programs. For a more fine-grained data comparison, refer to Table 2.

These three figures are all very different from each other. Starting with the TV channels genres classification of Fig. 3a, it is possible to conclude that the bulk of available TV channels refers to *Movies and Series*, *Sports*, and *Entertainment*. *News*, *Kids*, *Documentaries*, and *General* stations do not have a large expression on the full set of available TV channels.

However, when breaking this classification by the programs available for playback, as shown in Fig. 3b, the distribution changes significantly, particularly with respect to *Kids* programs, which become more relevant from a number of available programs perspective. Given that every TV channel is available on Catch-up TV over the same time-windows (7 days), these significant variations are due to the different program lengths on each TV channels. *Kids* TV



(a) Program Corpus Duration By Genre CDF.



(b) Program Requests Duration By Genre CDF.

Fig. 4 Program duration By genre

channels typically provide many programs but with shorter durations.

The most different distribution is that of Fig. 3c, which clearly shows which TV channel genres are favored by users. They overwhelmingly prefer content from generalist TV channels, followed by *Movies & Series*, and *Kids* content.

These conclusions are expected, given that generalist TV channels, often free-to-air, are usually the most popular on any country, and that *Movies & Series* and *Kids* content have a high replay value in virtue of not being time dependent, as opposed to *News* and *Sports* content which are shown to have a low replay value.

4.1.3 Content duration by genre

Taking into account the results of the previous genre distribution analysis, we conclude that channels with different genres exhibit different program duration distributions. Figure 4 provides a detailed look into this hypothesis by showing the program corpus and requests duration CDFs. The statistical information is presented in Table 1.

The program corpus results of Fig. 4a show that different genres exhibit distinct program duration CDFs. *Kids* programs are clear outliers with respect to the remaining categories, with most (>90 %) programs having a duration of less than 30 min.

Other genres have a more similar program duration CDF, but still exhibit genre-specific behaviors. For example, the *Movies And Series* and *Documentaries* genres show an almost perfect step-wise CDF with steep ascents for program durations at around 20–30 and 50–60 min. *General* and *Sports* genres are shown to have contents with longer average durations.

The comparison of these results with those of Fig. 4b, which focuses on what users actually request, shows a few key differences. Globally, users favor content with longer duration, as visible by the overall increase in program durations for each genre. A good example is *Kids* content, where a much stronger preference is shown for content with a duration between 20 and 30 min.

4.1.4 Video quality

As in the genre analyses, video quality information is limited for each individual TV station, and every program is assumed to have the same broadcasting quality as its parent TV channel: Standard Definition (SD) or High Definition (HD) for higher bitrate content.

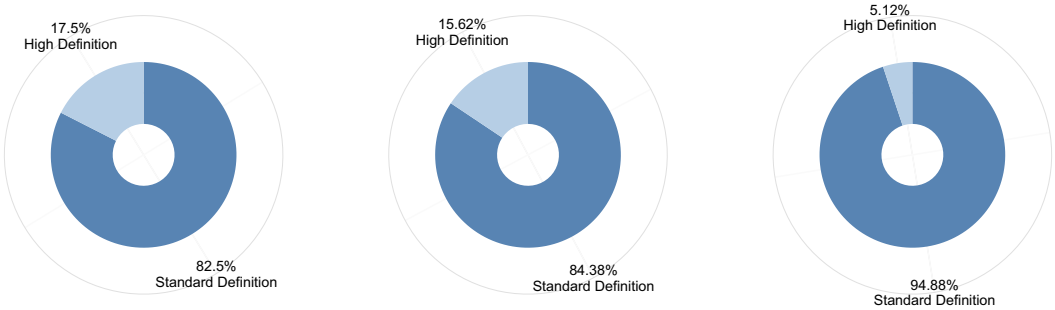
Figure 5a presents the distribution of SD and HD channels, Fig. 5b focuses on the same distribution, but for available programs, while Fig. 5c evaluates the video quality of playback requests.

The results of Fig. 5a, b show a similar video quality distribution; however, when considering Fig. 5c it is clear that the most requested programs are not streamed in HD, suggesting that users either prefer content quality over video quality, or they lack the technical requirements to request HD content, as may be the case on Digital Subscriber Line (DSL)-based Pay-TV subscribers whose bandwidth is constrained.

4.1.5 Fraction of program corpus requested

To understand if users take advantage of the entire content catalog, Fig. 6 compares the number of programs available at each day with the number of programs that have been requested at least once in that day.

The *Available* curve represents the total number of programs available at each day, which is 100 % of the daily program corpus, the *Requested* curve shows the percentage of programs requested at least once as a fraction of the total available programs, while the *Not Requested* curve represents the difference of the previous two curves. The shadowed curves represent the Locally Weighted Regression (LOESS) [8] smoothing with a 95 % Confidence Interval (CI) region, and are shown for readability reasons.



(a) TV Channels. **(b)** Program Corpus. **(c)** Program Requests.

Fig. 5 Content video quality distribution

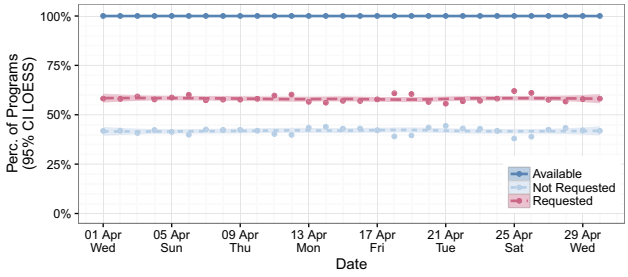


Fig. 6 Programs available vs. requested

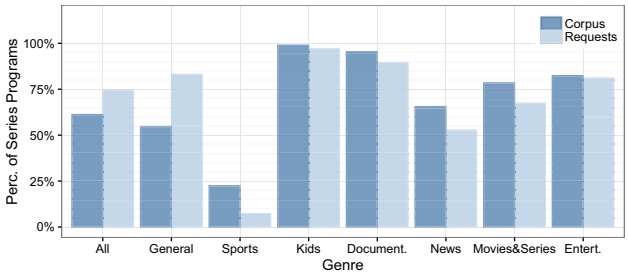


Fig. 7 Percentage of series programs by genre: available vs. requested

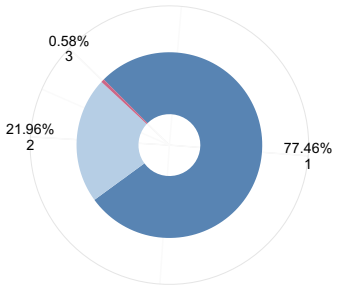


Fig. 8 Set-top boxes (STBs) per household

It is clear that users do not take advantage of the full catalog of Catch-up TV programs, as approximately 42 % of programs are never requested, as observed in Fig. 6 and Table 3. This observation suggests that these programs are highly irrelevant and are good candidates for deletion from the Catch-up TV catalog, or for having a very low priority in caching systems.

4.1.6 Serialized content

TV series are widespread in modern TV broadcasts and their popularity among consumers has risen to the point where a specific word exists for describing viewing marathons, i.e., *binge watching*.

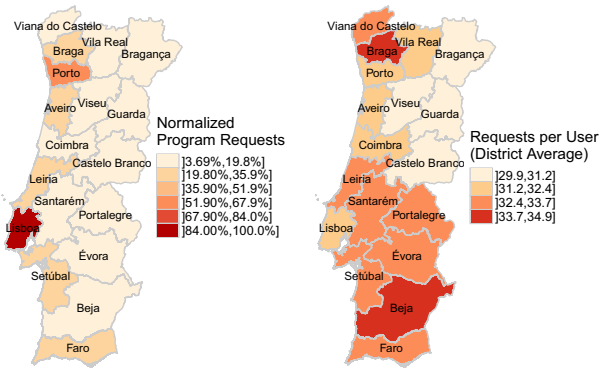
To understand both the offer and demand of TV series on this Catch-up TV service, the users' requests data are used to gauge the percentage of requested content that is serialized vs. non-serialized. Figure 7 presents a detailed breakdown on the percentage of serialized vs. non-serialized content by genre for the program corpus and requests (the *All* genre is used for data aggregation). Table 2 provides the summary statistics.

It is evident that some genres provide more serialized content than others. For *General*, *News*, and *Entertainment* channels, the percentage of requested serialized content is significantly higher than what is offered, while for the remaining genres the opposite is true.

Most programs broadcasted and requested on *Kids* and *Documentaries* channels are serialized.

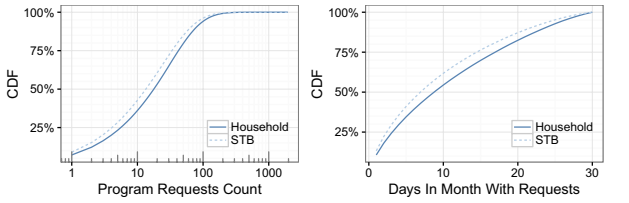
Sports channels broadcast approximately 17 % of their programs as TV series; however, only 3 % of their program requests refer to serialized content, thus, for this genre, users clearly prefer non-serialized content.

As a whole, considering the results of *All* genres, 75 % of program requests refer to serialized content vs. the corpus' 60 %. This metric demonstrates that users have a strong preference for TV series.



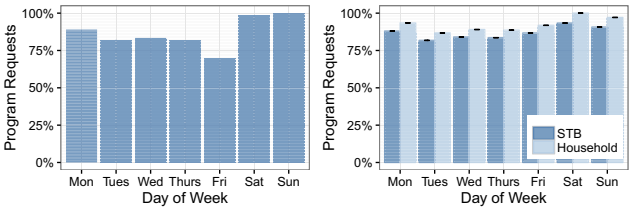
(a) Per District. **(b)** Per User per District.

Fig. 9 Geographical consumption overview



(a) Total Requests CDF. **(b)** Days With Requests CDF.

Fig. 10 Program requests frequency and intensity



(a) Program Requests Per Day of Week. **(b)** Program Requests Per Day of Week Per Type.

Fig. 11 Service usage per day of week

4.2 Service utilization

Having performed a comparison between several characteristics of the program corpus and users requests, this section provides additional insights on *where*, *when*, *how often*, and *how much* the service is utilized.

4.2.1 Number of STBs per household

Before delving into analyses considering different aggregation levels, either by household or STB, it is necessary to first consider the distribution of STBs per household presented in Fig. 8. Each household may have multiple STBs (up to 3), depending on the technical deployment limitations and users' subscription plans. The results only contemplate devices that have accessed the service at least once in the period under consideration.

It is evident that the large majority of users relied on a single STB to access the Catch-up TV service, 22 % used 2 different STBs, and only a very small fraction of users accessed the service on 3 different STBs.

Users mostly use the service on a single main STB, and the ensuing analyses reflect this reality.

4.2.2 Geographical distribution

Knowing *where* the service is mostly utilized is of utmost importance, as it enables service optimizations with respect to content placement and distribution strategies around the country, which are required to maintain a high user QoE. Figure 9 maps program requests onto districts of Portugal's main land.

Beginning with Fig. 9a, a high polarization is evident, with Lisboa and Porto districts capturing most program requests, which is not surprising given that they are Portugal's two largest cites, population-wise.

Figure 9b, which looks at the service usage intensity per user, reveals a completely different figure, and provides a better characterization of (average) users' behaviors according to the different districts. One of the least populated districts, Beja, exhibits the highest average number of program requests per user.

4.2.3 Frequency and intensity

Figure 10 addresses the question of how many program requests each STB and household performs per month, and also of how often the service is used. The detailed results are included in Table 4.

Figure 10a reveals that approximately 50 % of the households performed over 18 program requests over the course of the analysis period (1 month), while 25 % performed over 40 program requests. These results are consistent with those of Fig. 10b, which demonstrates that 50 % of the households use the service at least once every 3 days, while 25 % of them use it more intensively, i.e., approximately every other day. Naturally, given that each household comprises accesses from (potentially) multiple STBs, the households' CDF curves are shifted to the right of the STBs' CDF curves.

Taking into account the overall service utilization frequency and intensity, we conclude that this Catch-up TV service is regularly used and is a part of users' TV-watching routines.

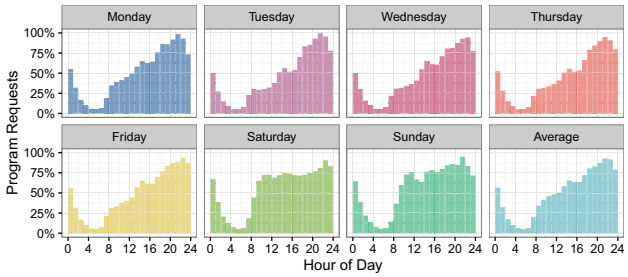


Fig. 12 Service usage: day of week and hour of day

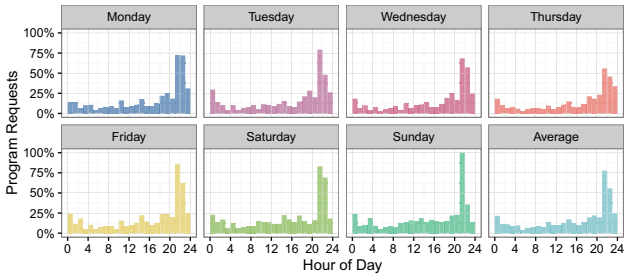


Fig. 13 Original airing: day of week and hour of day

4.2.4 *Usage by day of week*

While the previous analyses provide an overall overview on the service utilization, Fig. 11 provides a more detailed view on when the service is mostly utilized within each day of week.

Figure 11a shows that the Catch-up TV service exhibits a utilization spread out over the week, with particular emphasis on Saturdays, Sundays and Mondays (this is a trend in all weeks observed). While a higher utilization on the weekend is expected, as users tend to have more free time, the service utilization on Mondays is surprising, and may be due to users catching-up on programs that they missed on the weekend.

In spite of these differences, it is worth to point out that the most active day, *Sunday*, only has 31 % more requests than the day with less demand *Friday*, indicating that the service is widely utilized every day.

Figure 11b provides a different point-of-view on the weekly utilization data, by focusing on the number of program requests per user, to gauge the variations of utilization intensity. The high number of requests per household, when compared to the STB's, points to households where multiple STBs are used to access the service on the same day.

When comparing these results to those of Fig. 11a, a key difference stands out regarding *Friday*, which in spite of having the least overall program requests, it is also the weekday with the most avid users. This evaluation also shows that the service is more actively utilized on weekends and Mondays.

4.2.5 *Usage by hour of day*

Figure 12 presents the number of program requests per day of week and hour of day, to foster a better comprehension on *when* users request programs.

Beginning with a global examination on the characteristics of each individual plot, it is possible to conclude that users are less active on the late night hours, approximately from 02:00 to 07:00, and begin using the service more intensively from 08:00, up to a peak at around 21:00, regardless of the day of week.

The 02:00 to 07:00 interval corresponds to the normal sleeping hours, while the 20:00 to 23:00 interval matches the traditional prime-time.

On regular weekdays, the service utilization shows a continuous growth from 08:00 to the prime-time, while on weekends the service utilization is roughly constant throughout the day, with the exception of late night hours. This is as expected, as on weekends users are at home and watch Catch-up TV throughout the day.

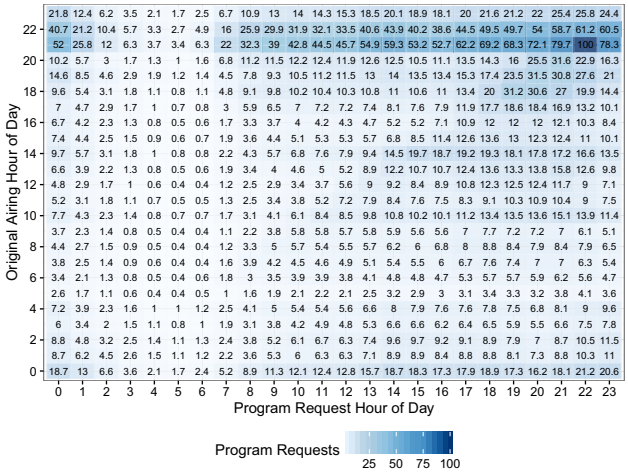
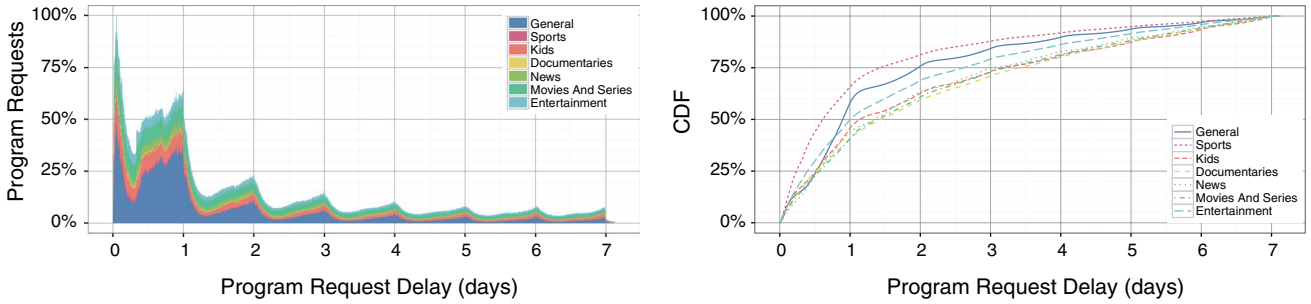


Fig. 14 Original airing hour of day vs. program request hour of day heat map

4.2.6 *Original airing time relevance*

This analysis is complementary to the previous one, in the sense that instead of concentrating on program request times, the key metric is the original broadcasting time, i.e, the day of week and time of day when the Catch-up TV program originally aired.

Figure 13 shows that content aired on prime-time is also the most popular Catch-up TV content, exceeding the popularity of content aired on other hours of day. Additionally,



(a) Total Requests vs. Request Delay.

(b) Total Requests vs. Request Delay CDF.

Fig. 15 Program requests decay

the results also show that prime-time content of Fridays, Saturdays and Sundays is more popular than prime-time content aired on other days.

4.2.7 Program request time vs. original airing time

A consolidated perspective of the previous two analyses that blends the program requests' hour of day evaluation of Fig. 12 with the program demand characterization of Fig. 13, is provided in Fig. 14, which presents a highly informative heat map that matches the program requests' hour of day with the original airing time hour of day.

By condensing a large amount of information into a single figure, it is possible to extract high-level insights on the relationship between the original broadcasting time and request time. The program requests data are normalized, thus ranging from 0 % (no requests) to 100 %—the maximum number of requests.

From the heat map, it is possible to identify a few key regions. Starting with high usage areas, the first corresponds to the area delimited by the original airing hour of day from 20:00 up to 23:59, and roughly matches the expected prime-time. Similarly, users are mostly active from 17:00 up to 23:59. Regions with fewer program requests are also easily identifiable, and correspond to programs originally aired between 05:00 and 09:59, and requested between 02:00 and 7:59.

Another interesting observation is that the most requested programs, at 22:00, originally aired 1 h earlier, at 21:00. In fact, a mild positive correlation is visible between the program request hour of day and original airing hour of day, which is shown as a diagonal region in the heat map and suggests that users also have a preference for programs that aired at their approximate service utilization hour of day (e.g., afternoon users requesting afternoon content).

4.2.8 Program requests decay

Catch-up TV enables an *anytime* approach to content consumption that removes the time constraints associated with watching linear TV. Considering this new degree of freedom, the question of whether users take advantage and watch Catch-up TV content without regard to how long ago it was originally transmitted arises; thus, the purpose of this analysis is to evaluate the evolution of content relevance as it ages and new content is added to the Catch-up TV catalog.

Each program is classified according to its channel genre, and the summary data are presented in Table 5.

The first observation of Fig. 15a is that peak program demand occurs within 1 day of the original broadcasting time. A decrease in demand is observed due to night periods throughout. Furthermore, some genres completely dominate the number of content requests, namely *General*, *Movies and Series*, and *Kids*. *Entertainment*, *Documentaries*, *Sports*, and *News* genres quickly become irrelevant after the first two days.

Figure 15b enables a per-genre evaluation on the evolution of program requests with time, and shows that they exhibit very different decay patterns. Two main types of genres are clearly visible: those whose relevance quickly fades with time, such as *Sports*, *General*, and to some extent *Entertainment*; and others, whose relevance does not decrease so significantly with time, as is the case of *Documentaries*, *Kids*, *News*, and *Movies and Series* genres.

Given the time sensitiveness of *News* programs, the results may seem odd; however, these TV channels are also known for hosting multiple sports and political debates, which might extend their overall time relevance.

As for the remaining genres, *Sports* and *General* have a high temporal relevance locality, while *Movies and Series*, *Documentaries*, and *Kids* programs do not typically exhibit any particular temporal importance, with the notable

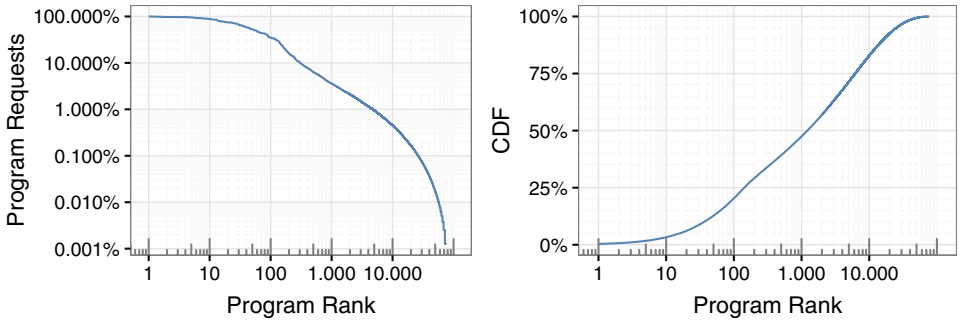


Fig. 16 Total requests per program

(a) Total Requests Per Program By Rank.

(b) Total Requests Per Program CDF.

exception of some high-impact TV series, such as Game of Thrones.

With more than 50 % of the total program requests on the first day, and 79 % after just 3 days, these results show that: on the one hand, the playback delay is a key factor on content popularity prediction, thus with the potential to be utilized in caching optimization algorithms; while on the other hand, increasing the Catch-up TV time window, from the current 7 days may not yield consumers any real benefit other than the psychological one of knowing that they have more content available, even if they will never watch it.

4.2.9 Program popularity

In addition to knowing *when* content is requested, it is also important to understand *which* programs have the highest demand, and how their popularity compares to the remaining programs. This is key to assess if Catch-up TV content exhibits either the *superstar* effect, or the *long-tail* effect [6, 10, 17].

To conduct this analysis, the programs were ranked according to their total number of requests, from the most popular (#1) to the least popular. Using the resulting data, Fig. 16 provides two complementary views on program popularity.

The results presented in Fig. 16a demonstrate a very large disparity on the amount of requests from popular vs. unpopular content, to the point of requiring logarithm scales on both axes. A very small subset of programs (∼ 300) exhibit a demand that is orders of magnitude larger than the rest.

These conclusions are reinforced in Fig. 16b, which shows that the top 1,000 programs are responsible for approximately 50 % of the total program requests, while the top 10,000 account for more than 80 %. The 70,000 least popular programs account for 23 % of the overall program requests; therefore, it is possible to conclude that Catch-up TV content consumption exhibits a predominant *superstar* effect.

4.3 Viewing sessions characterization

Following the service utilization analysis, this section aims to characterize the service utilization from the viewing sessions' perspective, i.e., by looking at user engagement periods instead of mere isolated program requests; thus, this higher level approach provides a holistic perspective on how the service is utilized.

Considering that the dataset available only contains information regarding the program requests' start times, it is assumed that users fully watch programs that they request, unless a subsequent request is made before the expected program ends, interrupting it.

In this back-to-back scenario, multiple subsequent program requests are consolidated into a single session.

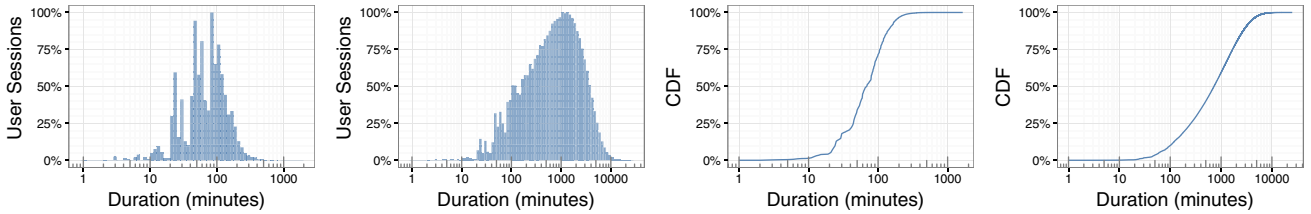
4.3.1 Session duration

The time spent by users in each individual viewing session is important to understand if users limit themselves to a single program, or if instead they watch several programs back-to-back, in a binge-watching session.

Figure 17a, c focuses on independent viewing sessions per STB, while Fig. 17b, d encompasses the cumulative duration of viewing sessions per STB for the full period under consideration. The summary statistical data are provided in Table 4.

A comparison between Fig. 17a, b reveals an apparent overlap for durations under approximately 100 min, indicating that some users limit themselves to watching one program; however, it is also possible to observe that many viewing sessions have a duration that is much larger than individual programs, thus pointing to the consecutive consumption of multiple programs. This conclusion is corroborated by the comparison of Fig. 17c with Fig. 17d, which shows an overall shift towards longer durations.

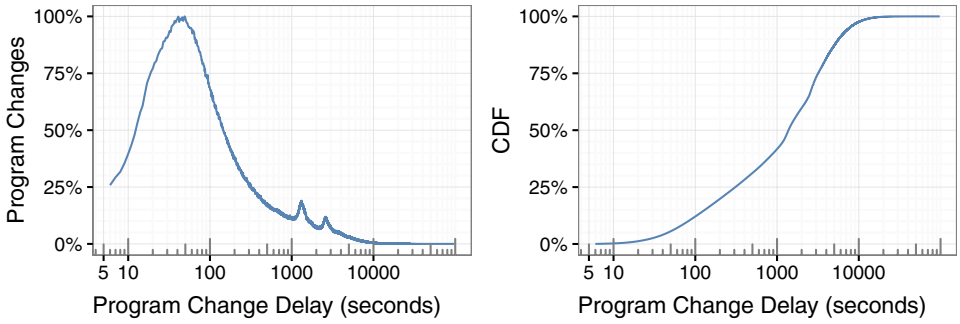
Figure 17b, d provides interesting information regarding the service usage. On the one hand, it is possible to observe that the histogram approximately follows a bell



(a) Individual Sessions' Duration Histogram. **(b)** Cumulative Sessions' Duration Histogram. **(c)** Individual Sessions' Duration CDF. **(d)** Cumulative Sessions' Duration CDF.

Fig. 17 Playback sessions' duration per STB

Fig. 18 Program change delay



(a) Program Changes vs. Program Change Delay. **(b)** Program Change Delay CDF.

curve, in spite of the three discernible lobes at lower durations. On the other hand, it is also possible to conclude that 50 % of users utilized the service for over 725 min in the considered period, which is significant and consistent with the total number of program requests analyzed in Fig. 10a.

4.3.2 Program change delay

Zapping and program changes are relevant aspects of users' behavior, which may significantly impact the design of OTT delivery architectures along with the users' service QoE.

Figure 18a, b shows two different viewpoints on program change delay, which computes the elapsed time between back-to-back programs requests. Considering that this analysis requires more than one program request, individual, non-overlapping program requests are not considered. The summary statistics are displayed in Table 4.

From Fig. 18a, three key regions are easily identifiable. The first region, ranging from 6 to 300 s, and a peak at approximately 50 s is consistent with zapping behaviors, while the secondary regions and corresponding peaks are consistent with sequential program requests and/or binge watching, as they occur at around 1300 s (22 min) and 2600 s (43 min), which are typical program durations, as seen in Fig. 2.

The results presented in Fig. 18b provide complementary behavior information, and show that 25 % of users change program after less than 309 s, while 75 % of them request another program within 53 min. This shows that prefetching after these initial 5 min might benefit the performance of CDNs.

4.3.3 Program requests in session

Previously, it was possible to empirically conclude that several sessions are composed of multiple program requests; however, not enough information was available to draw any conclusions regarding the number of consecutive program requests in a single viewing session. Figure 19 and Table 4 address this gap by providing information concerning the number of program requests per viewing session.

From Fig. 19a, it is clear that the most frequent number of program requests in a session is 1, i.e., that users select a single program and proceed to watch it, without selecting or zapping through additional content. This conclusion is supported by Fig. 19b where it is shown that 65 % of users select a single program in a viewing session, suggesting that they can easily find what they are looking for.

Figure 19a also shows that the likelihood of a user performing additional program requests decays abruptly, as corroborated by the fact 99 % of users request less than 9 programs in any given viewing session.

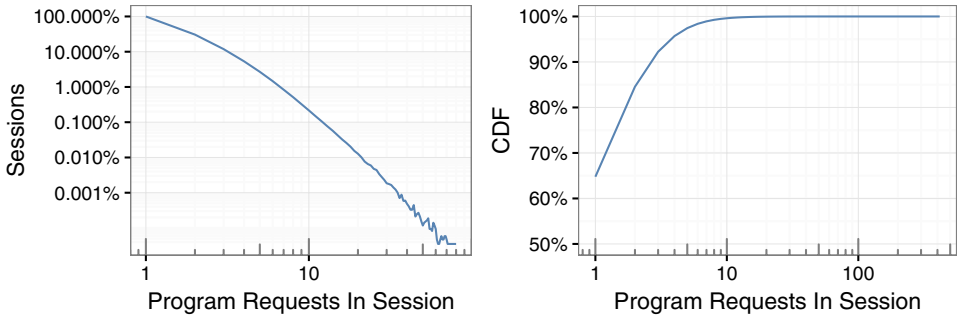


Fig. 19 Program requests per session

(a) Program Requests Per Session.

(b) Program Requests Per Session CDF.

4.3.4 Program settling time and requests

Before settling on a specific content, users skim through multiple programs, in a behavior known as program zapping. The characterization of zapping behavior is important to perform service optimizations, both from a technical and user experience standpoint.

Considering the results on viewing sessions' characterization already attained, specifically in Fig. 18, we delimit the zapping region as the one comprised with program changes with delays under 300 seconds (5 min). 1.6 % of the program corpus has a duration of less than this threshold, as is seen in Fig. 2c; however, only 0.22 % of program requests have this characteristic, hence we deem the potential analysis error of interpreting a normal program change as zapping acceptable.

To ensure an accurate analysis, data filtering was performed on the viewing sessions. First, a selection of viewing sessions with more than one program request was performed. Then, only the viewing sessions starting with a succession of program changes where each lasted less than the zapping threshold were kept. Finally, these sessions were further filtered to include only those that contained a program change that was not classified as zapping, i.e., the program on which the user settled for a long time.

This rigorous filtering aims to isolate clear zapping behaviors from others that might be dubious. Figure 20 starts by outlining the amount of zapping that takes place according to the content genre, in Fig. 20a, and proceeds to exploring two different dimensions on program settling: a total time perspective in Fig. 20b; and the number of programs changes in Fig. 20c. The statistical summary data are presented in Tables 2 and 4.

Inspecting Fig. 20a reveals that only a relatively small fraction of the viewing sessions, namely 13 %, exhibited any kind of zapping, leading to believe that most of the times users find what they want. In spite of this remark, it is also evident that the amount of zapping varies significantly between different content genres, and is mostly common on *News*, *Entertainment*, and *Sports* genres.

An initial examination of Fig. 20b reveals that 50 % of users require at most 115 s to find the content they are looking for. 95 % required less than the established zapping time of 300 s before settling on the final program. These settling times suggest that users take their time before changing between different programs, either because of limitations on the user interface, or because they require time to seek through the content to decide whether to watch it or not.

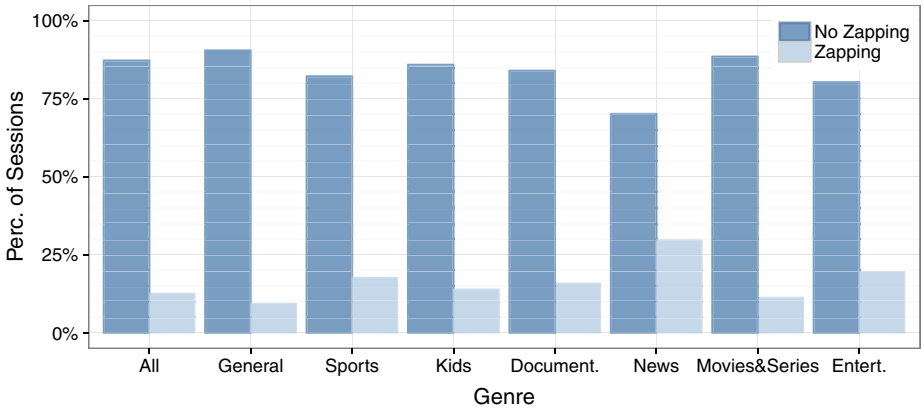
Additional complementary information is presented in Fig. 20c which addresses the program settling behavior from a number of zapping requests point-of-view. It is clearly shown that most users that "zap" (71 %) find the content that they are looking after a single zapping event, while 95 % require 3 or less program changes. This is key to plan and optimize content caching approaches, which can include a small amount of the content and prefetching.

4.4 Content delivery optimization

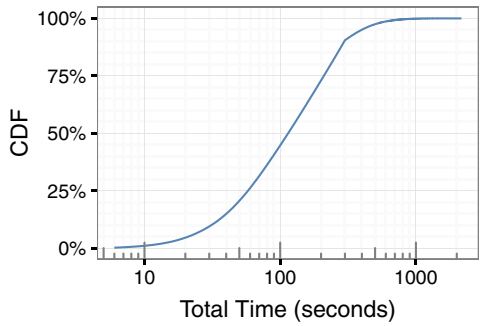
Even though the preceding analysis provides a deep insight into *how*, *when*, and *where* the service is utilized, with multiple perspectives on both content and users' characteristics, from a content delivery perspective, it is also important to understand how network traffic changes with time, and the potential gains achievable from smart caching and prefetching algorithms. These different viewpoints are essential to a properly planned and optimized CDN, in its various dimensions.

4.4.1 Bandwidth consumption

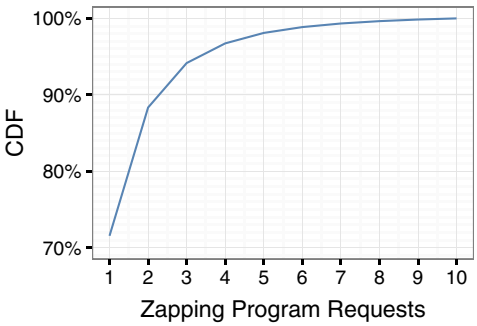
How much bandwidth is consumed and how it varies between peak and off-peak hours is determinant in network capacity and investment planning and to gauge the potential gains of network load distribution in time.



(a) Viewing Sessions Zapping Percentage vs. Genre.



(b) Program Settling Time CDF.



(c) Program Settling Number of Requests CDF.

Fig. 20 Program zapping, settling time and requests

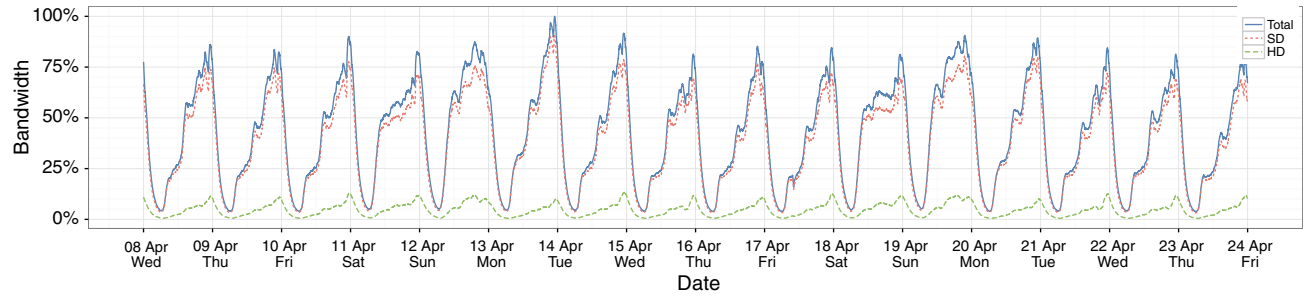
An accurate estimation on bandwidth consumption, which is a continuous measurement, must take into consideration not only the duration of each viewing session, but also the video quality of the requested programs. The approach followed in viewing sessions' characterization is also applied in this context, i.e., it is assumed that a given STB may only have one active program at a time, and that a new program request interrupts a previous one if it happens before the expected end of the active program.

Given that bandwidth measurements must take into account the active users at any given point in time, instead of the time of the program request, the viewing sessions data are expanded to provide information regarding the active programs at any given point in time, with a granularity of 1 min. This granularity is chosen as a compromise between accuracy and the computational effort required to generate the data. Given that only a few programs have a duration of less than 1 min, and that the programs' EPG-based durations have a resolution of 1 min, we deem this approximation satisfactory.

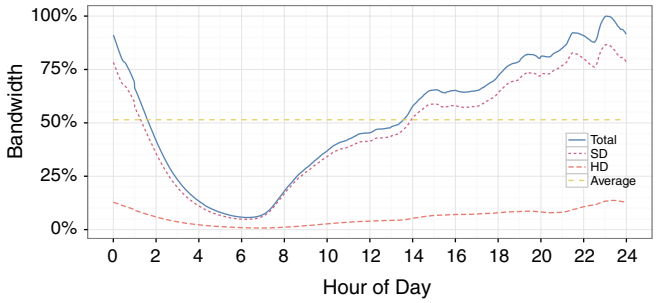
In addition to knowing which programs are active at each point in time, information is also collected regarding the video quality. HD content is streamed at 6 Mbps and requires exactly twice the streaming bandwidth of an SD content (3 Mbps).

Three different sub-figures are included in Fig. 21. The first Fig. 21a provides a high-level overview on the bandwidth consumption variation in the different days of month. Only 16 days—out of 30—are shown for readability purposes. Figure 21b examines the variation of bandwidth demand with the hour of day, by averaging the bandwidth consumption data of the different days. An alternative perspective of the same data is provided on the CDF of Fig. 21c.

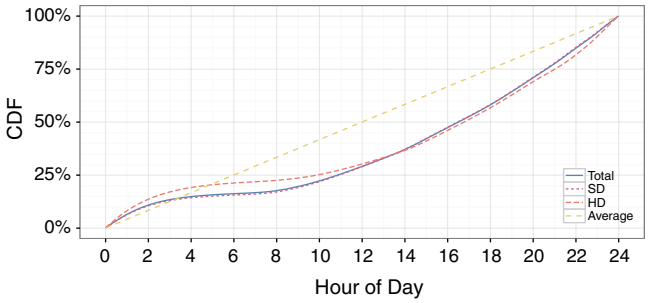
The explanation for the large gap in users watching HD and SD programs is threefold. The first reason is the lack of HD channels when compared to SD ones, as reflected in Fig. 5, where only 15.6 % of the programs available on Catch-up TV are HD. Second, the Catch-up TV user interface prioritizes SD over HD, which is an engineering design choice to reduce the overall bandwidth consumption as HD programs require twice as much bandwidth as their SD counterparts. Finally, because the vast majority of users are on DSL connections, with restrictions on bandwidth and amount of simultaneous video streams (the connection supports fewer HD streams than SD streams), users have an additional incentive to watch the SD versions in detriment of HD content.



(a) Bandwidth Consumption vs. Day of Month.



(b) Bandwidth Consumption vs. Hour of Day.



(c) Bandwidth Consumption vs. Hour of Day CDF.

Fig. 21 Service bandwidth consumption vs. time

4.4.2 Caching

This subsection is focused on metrics that provide insights on how cacheable Catch-up TV content is. To that end, an initial analysis of individual content popularity is conducted, to understand the theoretical limits of caching algorithms. Programs that are the most recurrently watched over their availability window present the best opportunity for caching improvements.

A study is conducted to determine how the cache storage requirements vary if they were to hold a given percentage of the most popular content. In this study, the programs available for request on each day are ranked according to their total number of requests.

While this approach does not take into consideration the impact of content locality, it does provide an overall perception on how cacheable Catch-up TV is.

The storage requirements are determined as a function of their duration and video quality. HD content, streamed at 6Mbps, requires twice the storage amount per unit of time than SD, which is streamed at 3 Mbps. On average, 15.8TB of storage space is required to hold the complete set of available Catch-up TV programs regarding the service's 7-day window.

Figure 22 presents the average results for the 30-day analysis with the 95 % CI as a shaded region.

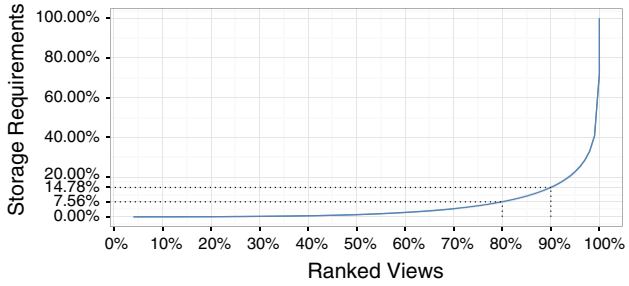


Fig. 22 Required storage size vs. top program requests

The first key observation is that the top 80 % of programs only require 7.56 % of the total storage requirements (∼1.2TB), which is not surprising considering the previous results regarding programs' popularity and demand presented in Figs. 6 and 16. The law of diminishing returns is clearly applicable, given than, for example, holding 90 % of the most popular programs in cache would require a twofold increase in total storage requirements (∼2.3TB).

Caches requiring 1.2TB of fast storage, such as RAM, are well within the reach of common servers. It seems evident, then, that a properly designed caching mechanism, integrated in a CDN, and aware of the particularities of Catch-up TV content demand, would be able to show a stellar caching performance.

5 Conclusion

Considering the objectives set forth in the introduction, which stress the importance of thoroughly understanding the Catch-up TV service usage to optimize the content delivery with benefits to operators and the consumers, it is clear that several highly valuable conclusions may be drawn from this study.

Regarding users' characterization with respect to their service access location and devices, it is shown that most requests originate from two main districts, Lisboa and Porto, and also that the vast majority of users (77 %) utilizes a single main STB to access the service.

Users have a preference for mostly *General*, *Kids*, and *Movies and Series* content when watching Catch-up TV, whereas the remaining genres have a lower overall expression, in spite of the high availability of content in the less popular genres. In addition to the content genre differences, users also seem to favor serialized content over one-off programs.

The exploration of the most popular programs' characteristics shows that they were originally prime-time programs whose popularity was reinforced in Catch-up TV, hence proving the *superstar* effect of Catch-up TV, as opposed to the *long-tail* one. Furthermore, the results also show that users are very active throughout the day, particularly on weekends.

When engaged with the service, 35 % of users do not limit themselves to a single program request, possibly due to *binge watching*. This affirmation is reinforced by the fact that while, on average, each household requests 32 programs per month, and uses the service 1 in 3 days, a significant portion (25 %) uses the service much more intensively, as shown in Fig. 10 and summary Table 4.

Given the importance of zapping on TV, these behaviors were also analyzed, and show that 13 % of the views exhibited some kind of zapping behavior, and that on average users require 2m:32s to find the content that they seek, which is much higher than on live TV. Even though part of this time may be due to operating the user interface, it also suggests that users skim through the content to decide if they want to watch it. This shows that prefetching techniques could provide significant optimization gains.

Notwithstanding the very large content catalog, the results show that more than 40 % of the catalog is not requested at least once, every day, thus indicating that significant storage gains might be achieved if they were identified and removed from the catalog.

These conclusions are complemented by the fact that Catch-up TV programs get over 75 % of their total views in the first 3 days after airing; therefore, implying that expanding the Catch-up TV window from 7 days up to 14 or 30 days would not provide a real benefit to users, in spite of the added costs to Pay-TV operators.

Understanding users' desires is important not only to enhance the services' technological aspects, but also to improve the service providers' relationship with their customers and reduce churn, by giving users seamless access to what they want to watch.

One the one hand, the intensive Catch-up TV service usage demonstrated in this work indicates that users crave control over what and when they watch TV, leading to a disruption on the existing editorial-control model of linear TV; while on the other hand, the fact that most Catch-up TV playbacks occur shortly after the original content airing evinces that Catch-up TV blurs, rather than breaks, the frontier between linear and nonlinear television.

From a more direct content delivery perspective, the service optimization analyses revealed the large differences between peak and off-peak bandwidth demand, which is problematic due to the average underutilization of network resources, which needs to be dimensioned to approximately two times the average streaming bandwidth to avoid network bottlenecks. One possibility to ameliorate this issue would be to preload content on the client devices on low-demand hours, i.e., late night hours, to "flatten" the bandwidth curve, reducing the chance of network-related issues on peak hours, and improving the overall service quality.

Continuing on the topic of service delivery optimization, the caching-oriented study clearly shows that a small fraction of the programs are responsible for the vast majority of program request, and that caching the most top 80 % would only require 7 % of the total corpus storage space.

In summary, all of these conclusions point to significant service improvement possibilities that can and should be used on next generation OTT multimedia CDNs to provide a better QoE to users, while simultaneously reducing Pay-TV operators costs.

6 Future work

Taking into consideration the previous conclusions, several additional challenges, questions, and opportunities arise that will be the target of future research work.

From a social and behavioral perspective, open challenges include an exhaustive comparison of the previous research works on IPTV content demand characterization, such as the ones presented in Sect. 2, to understand if, and how, users' behaviors are changing. The rising popularity of nonlinear services points to changes in the modern TV-watching paradigm, with impacts on users' lives and social interactions that should also be considered.

Future research, with the purpose of directly improving the performance of CDNs, will focus on creating predictive models able to forecast Catch-up TV content demand, whose scientific applications range from novel caching algorithms tailored towards efficient delivery of Catch-up TV content, prefetching mechanisms with the purpose of reducing the peak network bandwidth consumption, and dynamic resource provisioning algorithms capable of leveraging content demand patterns to optimize resource allocations. Together, these technological improvements are expected to enable cost-effective Catch-up TV services that are more efficient, while simultaneously improving users' QoE.

Acknowledgments The authors would like to thank Fausto Carvalho (Altice Labs, SA) and João Ferreira (MEO - Serviços de Comunicações e Multimédia, SA) for the key discussions and for providing the raw Catch-up TV consumption dataset.

Appendix: Statistical summary tables

See Tables 1, 2, 3, 4, 5.

Table 1 Program durations distribution summary statistics (minutes)

	Statistics		Quantiles				
	$\bar{x}$	σ_x	25 %	50 %	75 %	95 %	99 %
Program corpus							
All	48.46	42.15	24.00	40.00	60.00	100.00	200.00
General	73.21	78.28	32.00	50.00	83.00	165.00	390.00
Sports	63.91	53.13	30.00	50.00	90.00	120.00	280.00
Kids	19.22	15.10	12.00	15.00	24.00	30.00	60.00
Documentaries	42.07	16.68	24.00	46.00	60.00	60.00	88.00
News	48.05	38.28	25.00	31.00	60.00	99.40	185.00
Movies and series	62.52	32.93	45.00	48.00	90.00	111.00	152.00
Entertainment	41.96	38.17	28.00	30.00	53.00	60.00	238.00
Program requests							
All	66.40	42.31	30.00	56.00	90.00	120.00	195.00
General	82.86	43.31	51.00	80.00	100.00	145.00	210.00
Sports	89.93	51.96	54.00	90.00	120.00	150.00	255.00
Kids	25.79	13.66	21.00	25.00	30.00	30.00	80.00
Documentaries	45.32	16.46	41.00	47.00	50.00	60.00	90.00
News	64.10	39.03	30.00	60.00	90.00	113.00	180.00
Movies and series	73.10	34.57	47.00	57.00	101.00	120.00	152.00
Entertainment	49.64	33.00	29.00	45.00	56.00	70.00	210.00

Table 2 Per-genre summary statistics

	All (%)	General (%)	Sports (%)	Kids (%)	Docum. (%)	News (%)	M.&S. (%)	Enter. (%)
Content distribution								
TV channels	100	7.50	20.00	6.25	7.50	10.00	28.75	20.00
Program corpus	100	4.97	15.16	15.77	8.64	10.09	22.28	23.10
Program requests	100	41.94	2.27	16.89	1.78	7.58	20.99	8.55
Serialized content								
Corpus–non-series	38.80	45.29	87.06	1.05	5.94	67.35	36.58	40.96
Corpus–series	61.20	54.71	22.46	98.95	95.45	65.53	78.45	82.40
Requests–non-series	25.41	16.78	97.07	2.94	10.75	57.11	46.24	30.10
Requests–series	74.59	83.22	7.39	97.06	89.53	52.87	67.54	81.21
Zapping behavior								
Zapping	12.74	9.51	17.79	14.08	16.00	29.81	11.47	19.68
No zapping	87.26	90.49	82.21	85.92	84.00	70.19	88.53	80.32

Table 3 Programs available vs. requested summary Statistics

	Statistics		Quantiles				
	$\bar{x}$	σ_x	25 %	50 %	75 %	95 %	99 %
Number of programs							
Available	19305.63	98.61	19242.25	19289.50	19353.50	19455.40	19518.17
Not requested	8068.43	319.54	7944.25	8164.00	8307.25	8354.20	8566.13
Requested	11237.20	300.46	11026.75	11169.00	11476.25	11658.20	11864.88

Table 4 Viewing sessions and service usage summary statistics

	Statistics		Quantiles				
	$\bar{x}$	σ_x	25 %	50 %	75 %	95 %	99 %
Program requests							
Per household	31.97	40.42	6.00	18.00	43.00	78.00	187.00
Per STB	25.96	34.42	5.00	14.00	34.00	65.00	160.00
Days with Requests							
Per household	11.09	8.35	4.00	9.00	17.00	24.00	30.00
Per STB	9.65	7.86	3.00	7.00	15.00	22.00	29.00
Session data							
Individual duration (min)	81.02	54.93	45.00	69.00	107.00	151.00	266.00
Cumulative duration (min)	1228.54	1420.35	255.00	725.00	1685.00	3044.00	6550.81
Program requests	1.71	1.54	1.00	1.00	2.00	3.00	8.00
Change delay (min)	38.46	47.52	5.15	23.02	53.07	94.52	216.85
Zapping							
Settling time (min)	2.53	2.21	0.97	1.92	3.50	4.95	10.82
Settling changes	1.53	1.15	1.00	1.00	2.00	3.00	7.00

Table 5 Programs requests delay by genre summary statistics

	Statistics		Quantiles				
	$\bar{x}$	σ_x	25 %	50 %	75 %	95 %	99 %
Genre							
All	1d 19:07	1d 19:19	12:13	23:48	2d 16:04	4d 20:53	6d 20:14
General	1d 12:52	1d 14:44	12:34	21:31	1d 23:13	4d 00:38	6d 18:57
Sports	1d 05:11	1d 12:43	05:42	14:40	1d 10:24	3d 11:23	6d 15:29
Kids	2d 01:09	1d 23:39	12:09	1d 03:17	3d 03:51	5d 11:31	6d 22:05
Documentaries	2d 03:17	1d 22:54	12:30	1d 10:34	3d 10:00	5d 08:29	6d 20:42
News	2d 00:36	1d 21:13	13:14	1d 08:15	3d 00:49	5d 04:34	6d 20:16
Movies and series	2d 02:06	1d 21:52	13:35	1d 09:22	3d 05:14	5d 06:54	6d 20:13
Entertainment	1d 17:51	1d 18:49	09:27	1d 00:07	2d 14:40	4d 17:16	6d 19:00

References

1. Abrahamsson, H., Bjorkman, M.: Simulation of IPTV caching strategies. In: 2010 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), pp. 187–193. IEEE, Ottawa (2010). http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5588896
2. Abrahamsson, H., Bjorkman, M.: Caching for IPTV distribution with time-shift. In: 2013 International Conference on Computing, Networking and Communications (ICNC), pp. 916–921. IEEE, San Diego, CA (2013). doi:10.1109/ICCNC.2013.6504212. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6504212
3. Abrahamsson, H., Nordmark, M.: Program popularity and viewer behaviour in a large TV-on-demand system. In: Proceedings of the 2012 ACM conference on Internet measurement conference—IMC '12, p. 199. ACM Press, New York (2012). doi:10.1145/2398776.2398798. http://dl.acm.org/citation.cfm?doid=2398776.2398798

4. Abreu, J., Becker, V., Nogueira, J., Cardoso, B.: Time-shift services : a taxonomy and techno-business impacts of Catch-up TV. In: CENTERIS 2015—Conference on ENTERprise Information Systems, p. 6 (2015)
5. ANACOM: Subscription Television Service Statistical Information 2nd Quarter 2015. Tech. rep., ANACOM (2015). http://www.anacom.pt/streaming/STVS2quarter2015?contentId=1366508&field=ATTACHED_FILE. Accessed 12 2015
6. Beauvisage, T., Beuscart, J.S.: Audience dynamics of online catch up TV. In: Proceedings of the 21st international conference companion on World Wide Web—WWW '12 Companion, p. 461. ACM Press, New York (2012). doi:10.1145/2187980.2188077. URLhttp://dl.acm.org/citation.cfm?doid=2187980.2188077
7. Cha, M., Rodriguez, P., Crowcroft, J., Moon, S., Amatriain, X.: Watching television over an IP network. In: Proceedings of the 8th ACM SIGCOMM conference on Internet measurement conference—IMC '08, vol. 22, p. 71. ACM Press, New York (2008). doi:10.1145/1452520.1452529. http://portal.acm.org/citation.cfm?doid=1452520.1452529
8. Cleveland, W.S., Devlin, S.J.: Locally Weighted Regression: an approach to regression analysis by local fitting. J. Am. Stat. Assoc. **83**(403), 596 (1988). doi:10.2307/2289282. http://www.jstor.org/stable/2289282?origin=crossref
9. CNC: L économie de la télévision de rattrapage en 2014. Centre national du cinéma et de limage animée pp. 1–33 (2015). http://www.cnc.fr/web/fr/ressources/-/ressources/6592632
10. Elberse, A., Oberholzer-Gee, F.: Superstars and underdogs: an examination of the long-tail phenomenon in video sales (2007). http://www.people.hbs.edu/aelberse/papers/hbs_07-015
11. Famaey, J., Iterbeke, F., Wauters, T., De Turck, F.: Towards a predictive cache replacement strategy for multimedia content. J. Netw. Comp. Appl. **36**(1), 219–227 (2013). doi: 10.1016/j.jnca.2012.08.014
12. Gopalakrishnan, V., Jana, R., Ramakrishnan, K.K., Swayne, D.F., Vaishampayan, V.A.: Understanding couch potatoes. In: Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference—IMC '11, p. 225. ACM Press, New York (2011). doi:10.1145/2068816.2068838. http://doi.acm.org/10.1145/2068816.2068838
13. Lange, A., Benhamou, N., Joux, A., Gros, H., Guen, J.M.L.: Video on demand and catch-up tv in europe (2009). http://www.obs.coe.int/documents/205595/264625/VOD+2009+EN. Accessed 09 2015
14. Marshall, C., Venturini, F.: Bringing TV to Life, Issue III TV is all around you. Accenture (III), 16 (2012). https://www.accenture.com Accessed: 09–2015
15. Nencioni, G., Sastry, N., Chandaria, J., Crowcroft, J., Nishanth, S., Chandaria, J., Crowcroft, J.: Understanding and Decreasing the Network Footprint of Catch-up TV. In: Proceedings of the 22Nd International Conference on World Wide Web, p. 12. International World Wide Web Conferences Steering Committee, Rio de Janeiro, Brazil (2013). http://dl.acm.org/citation.cfm?id=2488388.2488472
16. Nielsen: The Digital Consumer (2014). http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2014 Reports/the-digital-consumer-report-feb-2014. Accessed: 09-2015
17. Rosen, S.: The economics of superstars. Am. Econ. Rev. **71**(5), 845–858 (1981). http://www.jstor.org/stable/1803469
18. Vanattenhoven, J., Geerts, D.: Broadcast, video-on-demand, and other ways to watch television content. In: Proceedings of the ACM International Conference on Interactive Experiences for TV and Online Video—TVX '15, pp. 73–82. ACM Press, New York (2015). doi:10.1145/2745197.2745208. http://dl.acm.org/citation.cfm?doid=2745197.2745208

Appendix D

QoE Assessment of HTTP Adaptive Video Streaming

QoE Assessment of HTTP Adaptive Video Streaming

André Salvador[1], Susana Sargento[1], João Nogueira[1,2]
[1] Instituto de Telecomunicações, University of Aveiro, Portugal
[2] Portugal Telecom Inovação e Sistemas, SA, Aveiro, Portugal
{andre.salvador, susana, joaonogueira}@ua.pt

***Abstract*—Quality of Experience (QoE) is a crucial characteristic of any multimedia service and must be accounted for during the service development and planning stages. Nonetheless, given its subjective nature, it is extremely difficult to use analytical methods to estimate the average Mean Opinion Score (MOS).**

Traditional progressive multimedia streaming is a well researched topic with respect to QoE, however, modern streaming services relying on advanced adaptive video streaming technologies, with specific characteristics, have yet to have an all-encompassing method for QoE estimation, as research work tend to focus on only one, or a small subset, of the technology's aspects, such as the impact of buffering events, bit-rate change frequency, or initial playout delay.

This paper proposes a model for determining the QoE estimate of a playback session of HTTP adaptive video streaming, encompassing its complete range of characteristics. Several key-metrics are extracted throughout the playback session, and then analyzed by an analytical method able to predict the consumers' QoE. A subjective QoE survey is conducted according to industry's best practices and recommendations in order to validate the proposed models. The obtained results show that both subjective and objective estimations produce similar results, hence validating the proposed model.

Keywords—QoE; quality of experience; adaptive streaming; Smooth Streaming; survey;

I. Introduction

The Internet is a fundamental commodity that most humanity has come to depend on. It has been growing in features, and complexity ever since it was created, and has evolved to support advanced multimedia services not initially foreseen.

Multimedia streaming has seen an outstanding growth in demand, fueled by ever increasing broadband speeds and community provided content. Streaming technologies, as opposed to *download-and-play* technologies, are characterized by the capability of a receiving device being able to consume the data while it is still being transferred, thus reducing the amount of storage required at the client to that of the playback buffer. Video streaming in particular requires a network connection with adequate performance especially in terms bandwidth, but also with respect to delay, depending on the application.

Regardless of the underlying technologies in multimedia streaming, a factor that has gained importance over that last years is that of Quality of Experience (QoE). QoE is a purely subjective metric, but it is so important that it can make or break the success of streaming service. It is heavily dependent on the underlying Quality of Service (QoS) parameters, but expands on it by taking advantage of human perceptions and focusing on the overall experience.

Adaptive HTTP streaming technologies aim to increase the users' QoE by embracing the natural variations of the underlying networks' performance, along with different terminal characteristics, while taking advantage of the ubiquitous HTTP infrastructure. The technology has gained traction with several implementations, including Microsoft's Smooth Streaming, Apple's HTTP Live Streaming (HLS), and the recently standardized MPEG-DASH. Given the characteristics of these adaptive streaming technologies, previous QoE estimation models do not directly apply, as they fail to encompass the new dynamics of a users' viewing session.

Previous works exist focused on QoE research in the context of adaptive streaming, however, they are restricted to the analysis of specific metrics, such as pause-intensity, or the impact of quality changes in QoE. An overall industry-calibrated approach has not, to best of the authors knowledge, been developed yet, and is thus the focus of this research work.

The remainder of this paper is organized as follows. Section II presents the related work, while Section III presents the proposed architecture to estimate the QoE. Section IV presents the implementation aspects of the proposed mechanism, while Section V describes the tested scenarios and presents the results. Finally, Section VI presents conclusions and points out future work.

II. Existing Approaches

QoE may be estimated through subjective and/or objective methods, however, it should be noted that any estimation is merely an approximation, as it varies from user to user [1].

Subjective methods [2] do not rely on technical characteristics given they are only based on human assessments of a video stream, and instead rely on a large number of surveys to have statistical significance. On the other hand, objective methods require concrete analytical metrics to classify the video stream and required subjective approaches for calibration; databases are usually made available with previously determined reference data, such as OPTICOM's Perceptual Evaluation of Video Quality (PEVq) [3].

Video streaming systems are complex because they can depend on many factors, such as codec, screen size, resolution, and others. For example, a low bit-rate video displaying on a 17" laptop client with a full High-definition (HD) screen will likely translate into a low QoE, but the exact same video on smart-phone client with a 3.5-inch screen will probably provide a higher QoE.

Figure 1 illustrates the interplay of different factors impacting the QoE, and shows that a proper determination of a QoE estimation is comprised of both subjective and objective evaluations, including technical factors, human biological factors, along with qualitative and quantitative analyses.

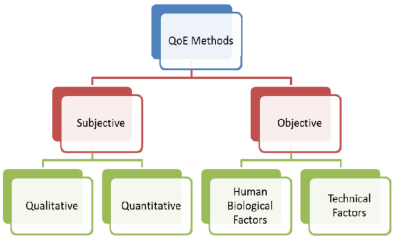


Fig. 1: QoE Assessment Methods [1].

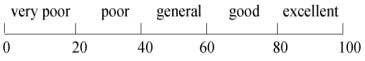


Fig. 2: One-hundred mark scale [4].

A. *Subjective methods*

Subjective assessment methods are commonly based on surveys, interviews, and statistical sampling of users to analyze their service perception [4]. There are two techniques for subjective studies (Figure 1):

Qualitative Techniques: Data is presented as verbal words and behavior. There are no results represented in numbers, but there are opinions, comments and questions. This technique can be advantageous in the so-called ratio of positive to negative comments (CCA - catalog, categorize, analyze). For example, a low bit-rate video shown in a small display may have good ratings from the users, while a video stream with a higher bit-rate displayed on a large screen may be poorly rated.

Quantitative Techniques: Data is presented as numbers and statistics. These studies can be performed in a laboratory environment to measure human feelings and perceptions. Typically, a questionnaire is applied with ratings on scales (Figure 2) for the data to be processed. To create scenarios with standard settings, an International Telecommunication Union (ITU)-T P.1202 (10/2012) recommendation exists which provides a parametric, non-intrusive bit-stream assessment of video media streaming quality [5].

B. *Objective methods*

The objective evaluation of video quality is based on mathematical models that estimate the results of subjective evaluations. The quality rankings are based on metrics that can be measured objectively and can be automatically evaluated by a computer program. There are two branches of objective assessment as follows:

QoS/Technology centric techniques [4]: In this particular case of objective methods, the video image quality is usually analyzed with Full-Reference (FR) or Reduced-Reference (RR) methods. FR methods require the original video for comparison, while the RR methods only require a set of the original video's characteristics at the client's terminal.

There are several ways to calculate the degradation of video with FR, as indicated below:

- Peak Signal-to-Noise ratio (PSNR) [6] computes the ratio of noise between the two images, transmitted and received.
- Structural Similarity (SSIM) [7] looks at differences by using the YUV color space.
- Video Quality model (VQ) [8] is a standard developed by the Institute for Telecommunications Sciences, National Telecommunication and Information Administration (TIS/NTIA), is standardized by the American National Standards Institute, and has been adopted in two ITU Recommendations, namely ITU-T J.144 and ITU-R BT.1683 IPPM. VQ model can produce similar video quality scores to Mean Opinion Scores (MOS) obtained in subjective tests, i.e., it is expected to reflect end user perception.

Work [9] is a case based in RR and its presents nonlinear model driven image quality scheme that is based on a Neural Network statistical estimator.

Physiological and cognitive techniques (Figure 1): These techniques examine users neurologically and cognitively by running electroencephalographies (EEGs), magnetoencephalographies (MEGs), functional magnetic resonance imagings (fMRIs), and near-infrared spectroscopies (NIRS). The tests examine the state of the user (sensors) and collect values for measuring QoE. This is the most expensive approach for objective QoE measurement [10].

C. *Approaches for determining QoE in Video Streaming*

Some works present objective methods to predict the QoE value, while other studies rely on subjective approaches. In this paper, the focus is on objective methods, given that they facilitate systematic and programmable implementations based solely on accessible metrics, such as video codec information, buffering events, and screen resolution, to name a few.

Several examples of works that investigate the effects of impact of adaptive video streaming in the user experience are presented, however, none analyses more than two different metrics simultaneously.

The work in [11] presents the QoE perceived from a subjective video quality assessment at two different bit-rates with Advanced Video Coding (AVC) and MPEG-4 Part 10, corrupted by typical wireless channel transmission errors. Albeit being relevant to understand and model the impact of channel transmission errors, the work is not directly applicable to adaptive HTTP video streaming, given that this technology relies on TCP transport connections, which are reliable in nature, and respond to packet losses with increased delay due to retransmissions. Additionally, pixel domain analysis are not feasible given that they are based on FR or RR models, which require reference data for quality comparison.

There are, however, works that can be applied to the context of adaptive HTTP stream, because they evaluate some of the characteristics readily available in adaptive streaming technologies such as: Frames per second (FPS), bit-rate, delay, and others. For example, the work in [12] refers to a study assessing the quality metrics in the context of freezing video, which evaluates the quality of user experience based on the

number of freezes. The user experience is affected by different metrics, such as the bit-rate and FPS.

A combination of objective and subjective approaches allows for QoE estimate based on both the video characteristics and the users' behavior. The work in [13] proposes a method to build a database of human mental data including the human losses by forgetfulness, where the main objective is to update/get the final value of the QoE, based on a model with time varying and lowest approximation error.

The Time-Varying Subjective Quality (TVSQ) report presents the effect of variation in quality over time based on a filter (Figure 4) [14]. The dynamic part of the model is only TVSQ. TVSQ simulates the human behavior under several video conditions, such as re-buffering events that, after an initial analysis, are stored in databases for future use.

The general model of Hammerstein-Wiener[15] (Figure 3) can be used to model a human memory filter solution, and presents filter that requires non-linear inputs and outputs. In this model, the specific quality ($q^{st}[t]$) is obtained in the first phase, which can be corrected ($\hat{q}[t]$) to approximate the value of MOS (Figure 3). This work notes that the first 15 seconds of human memory have more impact in the QoE.

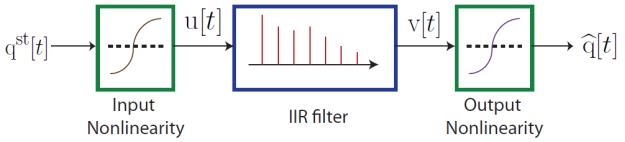


Fig. 3: Hammerstein-Wiener model for TVSQ prediction [14].

Figure 4 shows the IIR filter with 30 samples (f = 1Hz) that can simulate the Human Memory [16]. The response of the filter only considers the first 30 seconds as influencing to the MOS value.

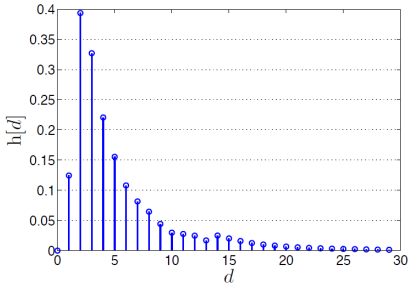


Fig. 4: The impulse response of the IIR filter in the first 30 seconds [14].

Other works corroborate the assumption that the impact of previous samples only affects the current experience during a limited amount of time. For example, the work in [16] postulates that the human brain presents two types of memories, short and long term. In this case, when the user watches a video, the probability of information to stand in the long term memory is very low, hence only the short term memory is considered. The user behavior and characteristics is very important in QoE estimation and the work in [13] presents a very good modeling solution.

Each of the analyzed works focus on particular aspects of QoE estimation, but none combines feasible quality estimations with the users' behavior in order to create a QoE estimation model that is both applicable to measurement softwares and is at the same time able to accurately depict the users' QoE. Our model proposal is presented in the following section.

III. QoE Assessment Architecture

In this section, the proposed model for accurate QoE prediction on adaptive HTTP streaming is presented. The goal is to devise a model usable by a streaming service provider, so that proper monitoring of the service performance, and its users' QoE, is performed.

Because the overall experience of a video streaming session up to a given instant is influenced by the previous instants, the model needs to consider a memory effect over the elapsed period.

The proposed algorithm may be decomposed into two phases, illustrated in the building blocks of figure 5.

A first one classifies individual video chunks regardless of others. It considers the video codec information, the client's terminal characteristics, and the network's QoS parameters in order to establish a baseline MOS for each individual video chunk, and is calibrated against PEVq.

The second phase builds on the basic classification of video chunks performed on phase 1, with respect to their individual MOS estimates, and considers the impact of the previously reproduced chunks in the current MOS, thus emulating the human memory effect.

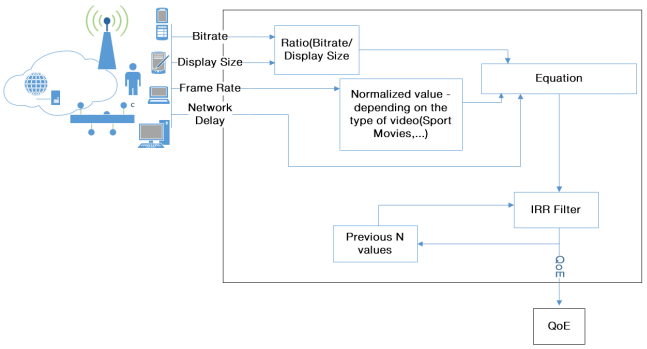


Fig. 5: Adaptive HTTP Video Streaming QoE Estimation Architecture.

A. Video Chunk Scoring

Initially, an objective assessment of the video chunks is performed using features that are independent over the time, such as the bit-rate and the FPS of a specific chunk, from a particular quality level - adaptive streaming technologies provide different quality levels, or representations, for the exact same content.

This multi-parameter assessment is performed by carefully and separately analyzing the impact of each particular metric in the MOS estimate. The impact of the variation of each parameter is determined through the use of previously calibrated tools that contain databases of the MOS values, such as PEVq, which provides reliable MOS values, obtained according to ITU recommendations [3].

The selection of metrics is essential to the design of the model given that, for practical reasons, it can only use metrics that are obtainable in the context of video players. Most related studies in this area consider only one or two metrics, as the increase on the number of metrics also increases the complexity of the algorithms, in spite of a reduction in the forecast error. As description of methods used in the video assessment process ensues.

Bitrate: The bit-rate metric contributes largely to the quality of user experience. In adaptive HTTP streaming, the videos are encoded with a constant bit-rate so that each video chunk (of given quality level), has an approximately equal wire size (in bytes). Constant bit-rate encoding is a problem with videos that have highly dynamic scenes, such as sports, because it results in lower compression gains. The encoded bit-rates of adaptive HTTP streaming videos, typically range from 250 Kbps to 3 Mbps.

Frames per Second: The maximum rendered FPS are usually a limitation of the devices' performance, albeit it is also upper limited by the quality of the video chunk, which depending on may reduce the number of frames per second, in exchange of a higher resolution (for a specific average bit-rate). A drop in FPS is most of the times immediately evident to video consumers.

Rebuffering: Rebuffering is characterized by the amount of times elapsed while a player waits for the download of a new chunk after suffering a buffer under-run. This has a crucial impact on QoE and significantly effects the user experience it is lasts over a couple of seconds.

Screen resolution [ratio]: The relation between the screen resolution and the video track is relevant to estimate the users' QoE. Thus, it is possible to differentiate the QoE from a device with a small screen and a device with a large screen. The impact of the screen size is heavily dependent on the users' viewing distance (this effect was studied in work [18]), so it is required that the stream resolution is within range of the devices' screen resolution.

B. Human Memory Filter

The previously detailed video chunk scoring approach is based on metrics that do not depend on time, however, in practice the user experience does, as the human memory plays a role in quality of experience perception.

Take as an example a situation where the user is watches a video comprised of two video chunks, with different qualities. If the user first watches the chunk with the highest quality and then the one with the lowest quality, his perception of QoE will be lower than if he had first watched the low quality chunk and then the higher quality one, even if in practice the average chunk quality is the same.

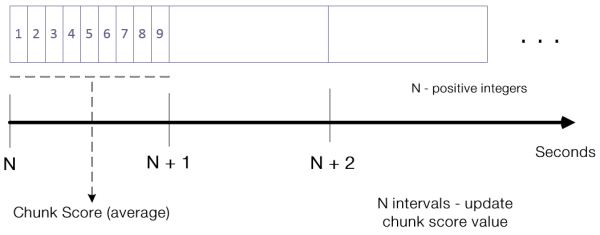


Fig. 6: Method proposed - Sampling Frequency.

This example illustrates the need of a memory filter able to replicate the impact of past experiences in the current evaluation. The proposed model applies a filter with a sampling frequency of 1Hz, which updates the current QoE estimation considering the previously displayed chunks.

IV. Implementation

The proposed method to estimate the QoE described in the previous section results in a two phase architecture: the first related to individual chunk scoring and a second relating to the perception of chunk sequences by the user.

In the chunk scoring phase, the sampling frequency is crucial to the analysis. Typical chunks contain 2 seconds of video, but re-buffering events may occur in smaller intervals. The human perception is defined by the eyes, to which 42 ms of the sampling is an acceptable value; thus, the chunk score can be done at 100 milliseconds intervals(figure 6), so that in each second 10 quality samples at produced that may be used in the second phase, that of the human memory filter.

An equation is proposed that relates a complete set of technical metrics, previously described, and then outputs a video quality estimation. Equation 1 relates the metrics behavior, and the calibrated equation is obtained by determining the values of *v1, v2, v3, v4, v5, v6, v7, v8*.

The calibration of the equation is performed using MOS values obtained by a variation of the video characteristics. In the first phase, the re-buffering and the screen size metrics were not considered ($Rebuffering = 0$ & $Screen_{size} = 1$) in the process to calibrate the other metrics in the equation. The re-buffering and the screen size were considered in the process calibration in a later phase.

$$
\begin{array}{lll}
v_1 = 2.038 & v_2 = 1.027 & v_3 = 1.42^{-6} \\
v_4 = 0.3031 & v_5 = 3.064 & v_6 = 0.5407 \\
v_7 = 0.05652 & v_8 = 1.756 &
\end{array}
$$

$$
\begin{aligned}
Score_{chunks} = & v_2 \arctan\left(Bitrate \times v_3\right) \times \log\left(v_4 \times FPS\right) \\
& - \log\left(v_5 \times Rebuffering + v_6\right) \\
& - \log\left(v_7 \times Screen_{size} + v_8\right) + v_1, \qquad (1)
\end{aligned}
$$

This equation outputs a MOS value with respect to the characteristics of the video, and there is the need to measure the evaluation of user experience in a streaming session.

Thus, this equation calculates a chunk value that will later be upgraded with a human memory filter.

Figure 4 shows a filter behavior with the influence of the first 30 seconds in the human memory.

Initially, in the evaluation of the video stream, the method starts the playback without previous values. There is no history of the session information when the video stream is started, so the lack of previous values is initially a problem, since the last 30 samples (fa = 1Hz) are necessary for the method to apply the filter of human memory.

The proposed solution adjusts the influence of the samples to 100 percent; for example, the first evaluation (starts at 0 seconds) of the chunk depends only on the current value. The next evaluation (starts at 2 seconds) depends only on the previous and current value. These values are adjusted to 100 percent, with the previous value influencing 63.75 % , and the actual value influencing 36.25.

In this architecture, it is necessary to verify the influence of the human memory filter through real scenarios that use subjective methods (surveys), which will be presented in the next section.

V. Results

This section presents the evaluation results of the proposed model, in order to validate its performance. A simulation scenario used for the evaluation is outlined, followed by a description and analysis of the experimental scenarios.

A. Simulation Scenario

In the simulation scenario, the proposed model for QoE estimation is compared with PEVq-calibrated results for individual chunks. This scenario is used to verify the impact of memory in a streaming session.

Figure 7 illustrates the time line of a streaming session where network conditions vary significantly and some buffering events occur. Up to 10 seconds into the streaming session, the video quality rises with the rise of chunk quality, but when a congested network reduces the available bandwidth and causes playback interruptions, the estimated QoE is heavily impacted. This scenario may occur, for example, when a user starts watching television on a tablet and then goes into another room where the wireless network has a weak signal [19].

The outcome of this simulation is presented on figure 8, where both the individual chunks' expected QoE is presented (in rectangular shapes), and the outcome of the proposed algorithm is displayed as curve with discrete estimates of the QoE value.

A disparity between the assessments is evident. Whereas the individual chunk quality immediately produces a particular QoE, and maintains it while the chunk bit-rate does not change, the proposed model is both dynamic and more conservative in the sense that it considers past experiences, thus not showing instant QoE variations, and outputting lower QoE values representing the negative impact of buffering events and quality transitions.

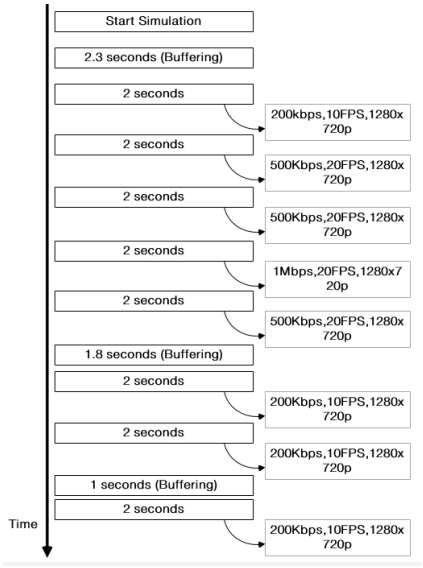


Fig. 7: Scenario - Video session time line.

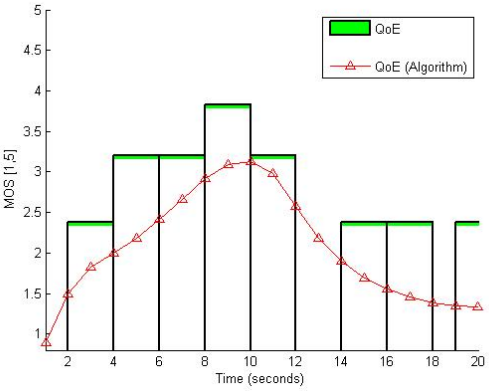


Fig. 8: Scenario - QoE variation with time (s).

B. Experimental Scenarios

Given that QoE is a subjective concept, a subjective approach to determine MOS values is required so that the proposed model may be benchmarked against real-life results.

In the experimental scenarios, a survey was conducted using real users to assess their quality of experience when watching variations of 2 reference videos: an animation one, and a sports one. The videos were made available in a set of 20 videos streams, with variations in the quality levels usable by the users' adaptive player (different sets of bit-rates per video stream).

ITU recommends that questionnaires should have at least 50 responses in order to have enough confidence in the results, hence we considered 64 users assessing the quality of the 20 video streams available on a web page. Each video stream is classifiable with a MOS score, ranging from 1 to 5. In practice, however, it is difficult to get an average MOS higher than 4.5 or lower than 1.5, because not everyone classifies their experience with the extreme values of 5 or 1.

Figure 9 shows the results of the video qualities questionnaires, indeed demonstrating that the users' MOS estimate does not present values near the extremes (MOS equal 5 or 1).

The results show that MOS estimates produced by the survey are in line with the estimates provided by the QoE model, especially in the case of animation video streams (scenarios 15 to 20). In scenarios 1 to 15 the reference is the sports video, and the QoE model does not perform as good as it does in the animation video. This is likely a general effect of sports videos, whose picture quality is usually harder to estimate due to fast moving scenes.

Overall, it is possible to conclude that the proposed model is able to closely track the subjective results, and does not present results near the extremes. As a side note, scenarios 1 and 11 are the same but are separated by 10 intermediate scenarios. It is expected that when a user is viewing scenario 11 he/she does not remember scenario 1, thus it is used as a user coherence validation test.

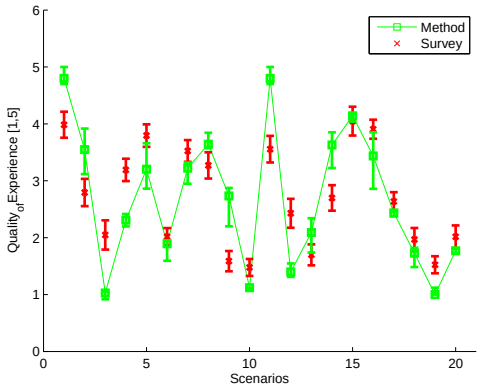


Fig. 9: Survey with 20 Scenarios.

VI. Conclusion & Future Work

This paper demonstrates a mechanism to estimate the QoE of an adaptive HTTP video streaming service aiming to simulate human video scoring behavior. In order to validate the performance of the developed method, both objective and subjective tests are executed.

In the objective tests, a comparison is performed between the proposed method and PEVq over different network quality scenarios and metrics, such as bit-rate, FPS, re-buffering time and screen size. Furthermore, the maximum deviation was 0.19 in the MOS scale (ranging from 1 to 5). Additional tests evaluating the impact of the video content in the proposed algorithm results lead to the conclusion that the confidence interval is not exceeded in most of the cases, thus demonstrating that the video content does not impact significantly the QoE estimate.

In the subjective assessments, a questionnaire is designed to recreate test scenarios comparable with the ones performed by the objective MOS estimation approach. In the animation video scenario, the results were an almost perfect match to the objective estimate, but the sports video led to small discrepancies caused by the lack of identical submissions.

The all-encompassing approach taken in development of the proposed model enhances the current state of the art by demonstrating the incorporation of the key characteristics of adaptive HTTP streaming in the estimation of the users' QoE. These models will be incorporated in a service provider's QoE probing system.

References

[1] K. ur Rehman Laghari, O. Issa, F. Speranza, and T. Falk, "Quality-of-experience perception for video streaming services: Preliminary subjective and objective results," in *Signal Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, Dec 2012, pp. 1–9.

[2] A. Moorthy, L. K. Choi, A. Bovik, and G. de Veciana, "Video quality assessment on mobile devices: Subjective, behavioral and objective studies," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 6, no. 6, pp. 652–671, Oct 2012.

[3] G. OPTICOM GmbH, "Pevq - advanced perceptual evalutoin of video quality," http://www.pevq.com/, 2005, [Online; accessed 4-June-2014].

[4] D. Li and M. Cai, "A video quality-estimation model for streaming media services based on human visual system," in *Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on*, Dec 2009, pp. 1–4.

[5] M. P. E. Group, "Recommendation p.1202 (10/12)," http://www.itu.int/rec/T-REC-P.1202-201210-I/en, Approved in 2012-10, [Online; accessed 10-June-2014].

[6] T. Oelbaum and K. Diepold, "Building a reduced reference video quality metric with very low overhead using multivariate data analysis," 2007.

[7] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, April 2004.

[8] E. P. Ong, M. H. Loke, W. Lin, Z. Lu, and S. Yao, "Video quality metrics - an analysis for low bit rate videos," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 1, April 2007, pp. I–889–I–892.

[9] D. Bordevic, D. Kukolj, M. Pokric, and I. Ostojic, "Image quality assessment using reduced-reference nonlinear model," in *Intelligent Systems and Informatics (SISY), 2013 IEEE 11th International Symposium on*, Sept 2013, pp. 167–170.

[10] S. Arndt, J. Antons, R. Schleicher, S. Moller, and G. Curio, "Perception of low-quality videos analyzed by means of electroencephalography," in *Quality of Multimedia Experience (QoMEX), 2012 Fourth International Workshop on*, July 2012, pp. 284–289.

[11] A. Floris, L. Atzori, G. Ginesu, and D. Giusto, "Qoe assessment of multimedia video consumption on tablet devices," in *Globecom Workshops (GC Wkshps), 2012 IEEE*, Dec 2012, pp. 1329–1334.

[12] T. Minhas, M. Shahid, A. Rossholm, B. Lovstrom, H.-J. Zepernick, and M. Fiedler, "Assessment of the rating performance of itu-t recommended video quality metrics in the context of video freezes," in *Telecommunication Networks and Applications Conference (ATNAC), 2013 Australasian*, Nov 2013, pp. 207–212.

[13] L. Hong, S. Zheng, Y. Zhang, and L. Tan, "Parallel c-means algorithm in human memory meta database search mechanism," in *Web Information Systems and Mining, 2009. WISM 2009. International Conference on*, Nov 2009, pp. 133–136.

[14] C. Chen, L. K. Choi, G. de Veciana, C. Caramanis, R. W. Heath, and A. C. Bovik, "Modeling the Time;Varying Subjective Quality of HTTP Video Streams With Rate Adaptations," pp. 2206–2221, 2014.

[15] A. Wills, T. Schon, L. Ljung, and B. Ninness, "Identification of hammerstein-wiener models," *Automatica*, vol. 1, no. 1, pp. 1–14, dec 2011.

[16] P. Hong-jun, Y. Xiao-qiu, Q. Chang-song, and C. Hong-tao, "A category theory model for learning and memory of the human brain," in *Digital Manufacturing and Automation (ICDMA), 2010 International Conference on*, vol. 1, Dec 2010, pp. 11–14.

[17] IQinVision, "Nine key considerations when selecting a video compression algorithm," http://www.iqeye.com/iqeye/H.264_Considerations.pdf, •, [Online; accessed 3-June-2014].

[18] K. Sakamoto, S. Aoyama, S. Asahara, K. Yamashita, and A. Okada, "Evaluation of viewing distance vs. tv size on visual fatigue in a home viewing environment," in *Consumer Electronics, 2009. ICCE '09. Digest of Technical Papers International Conference on*, Jan 2009.

[19] G. Lui, T. Gallagher, B. Li, A. G. Dempster, and C. Rizos, "Differences in RSSI readings made by different Wi-Fi chipsets: A limitation of WLAN localization," in *2011 International Conference on Localization and GNSS, ICL-GNSS 2011*, 2011, pp. 53–57.

Appendix E

Catch-up TV Forecasting : Enabling Next-Generation Over-The-Top Multimedia TV Services.

Manuscript submitted to the *Springer's Journal of Multimedia Tools and Applications.*

Multimedia Tools and Applications manuscript No.
(will be inserted by the editor)

Catch-up TV Forecasting: Enabling Next-Generation Over-The-Top Multimedia TV Services.

João Nogueira · Lucas Guardalben · Bernardo Cardoso · Susana Sargento

Received: date / Accepted: date

Abstract Due to recent developments in Over-The-Top (OTT) technologies, Pay-TV operators have begun a migration process of managed IP Television (IPTV) services to more appealing OTT approaches. In these scenarios, being able to predict when and what resources will be necessary at any given point is crucial to a high-quality, efficient, and cost-effective operation, especially when dealing with the dynamic and resource-intensive requirements of IPTV multimedia services.

To evaluate the advantages of demand forecasting for efficient Catch-up TV delivery on OTT scenarios, this research work explores several classes of machine learning models regarding their accuracy, computational requirement trade-offs, and deployability. The training process relies on a dataset comprised of Catch-up TV usage logs acquired from an IPTV operator's live production service containing over 1 million subscribers. A predictive and dynamic resource provisioning approach is proposed and evaluated in terms of bandwidth and storage savings.

Results demonstrate that forecasting Catch-up TV demand is practical, suitable for integration in OTT solutions, and useful in improving efficiency, with benefits to operators and consumers. Significant savings in bandwidth and storage are shown to be achievable, enabling green and cost-effective resource usage.

Keywords Catch-up TV · IPTV · OTT · Multimedia · Forecasting

João Nogueira
Altice Labs, SA, Aveiro, Portugal
University of Aveiro, Aveiro, Portugal
E-mail: joaonogueira@ua.pt

Lucas Guardalben
Instituto de Telecomunicações, Aveiro, Portugal
University of Aveiro, Aveiro, Portugal
E-mail: guardalben@ua.pt

Bernardo Cardoso
Altice Labs, SA, Aveiro, Portugal
E-mail: bernardo@alticelabs.com

Susana Sargento
Instituto de Telecomunicações, Aveiro, Portugal
University of Aveiro, Aveiro, Portugal
E-mail: susana@ua.pt

1 Introduction

The rise in popularity of Catch-up TV services in the past decade has been accompanied by a continuous investment in walled garden managed delivery infrastructures, which support most Pay-TV operators' value-added services. This technological approach contrasts with the development and stellar growth of OTT multimedia, along with its widely known advantages regarding client device heterogeneity, scalability, and reduced Capital Expenditures (CAPEX) and Operational Expenditures (OPEX).

In the face of strong Pay-TV market competition and technological advances, Pay-TV operators are starting a natural evolutionary process towards OTT delivery of multimedia services, facilitating the *anytime-anywhere* promise of a convergent solution, while simultaneously relieving the economical burden of these storage and bandwidth intensive services [6].

A keystone in OTT multimedia services, which, if not properly accounted for, severely limits the systems' scalability and end-users' Quality-of-Experience (QoE), is Content Delivery Network (CDN) infrastructure optimization in its many aspects, ranging from caching optimization, bandwidth reservations, Point-of-Presence (PoP) location, and elastic resource provisioning to cope with varying demand [26]. While static optimization is possible, by thoroughly analyzing past demand data, it is error prone and subject to human-error. A more interesting scenario with potentially higher efficiency gains is that of autonomic and dynamic CDN optimization, capable of providing better resource usage, lower costs, and power consumption; however, this dynamic approach is rife with difficulties and is accompanied by a crucial obstacle: the need to accurately forecast demand in a practical time frame.

This necessity is addressed in this research work, which creates and evaluates forecasting models suitable for being employed as part of a solution for cloud resource orchestration in CDNs [26, 24], following a step-by-step approach to ensure clear, reproducible, and sound results that may be used in subsequent research efforts and applied to existing or new CDNs. In order to create, assess, and propose feasible and accurate forecasting models for Catch-up TV content consumption, Catch-up TV usage logs are obtained from a leading Pay-TV operator's production service, containing over 22 million requests over the span of 1 month.

A predictive and dynamic resource provisioning approach is proposed and evaluated in terms of bandwidth and storage savings. The attained results show that the forecasting models are able to produce accurate bandwidth and storage requirements forecasts, which may be used to achieve considerable power and cost savings.

The contributions of this paper are as follows:

- Definition of a step-by-step approach for performing Catch-up TV demand forecasts;
- Proposing the Training Average Scaled Error (TASE) metric, a new approach for comparing the performance of different predictive algorithms;
- Benchmarking several classes of machine learning algorithms in the context of Catch-up TV forecasts;
- Evaluation of bandwidth and storage requirements using the predictive models' outputs.

This study starts by performing an initial literature review on Section 2, after which, on Section 3, a detailed analysis and feature engineering is conducted on the available dataset. Next, Section 4 describes the forecasting methodology along with the necessary data transformations and feature selection. Section 5 describes the models' training, whose results are presented on Section 6. The concluding remarks and future work are presented on Section 7.

2 Related Work

Catch-up TV is a key differentiating feature in modern Pay-TV services, whose popularity often surpasses other advanced time-shifting features such as Video-on-Demand (VoD) and Digital Video Recorder (DVR) [1, 21].

As a consequence of its popularity, Catch-up TV imposes a severe strain on the delivery infrastructures, and has motivated researchers to tackle modeling and optimization challenges with the purpose of improving current delivery services and architectures.

From a modeling perspective, public and private datasets have been leveraged to draw conclusions regarding users' behaviors, which are then used to extrapolate impacts on the corresponding delivery services.

In [23], the authors rely on a large Catch-up TV consumption dataset, from a production IPTV service, to characterize multiple aspects of users' behaviors, including the duration of viewing sessions, zapping frequency, program settling delay, genre preferences, and program popularity analysis, to name a few. This behavioral study is supplemented by an analysis demonstrating potential bandwidth-saving gains by using caches of a small fraction of the overall content available. The statistical analyses presented in this study clearly support the existence of popularity and consumption patterns, depending not only on the time of day and week at which the service is used, but also on content characteristics such as its original airing time, date, and genre. They provide an indication that forecasting algorithms may be able to anticipate demand and provide knowledge that may be used by next generation CDNs to operate more efficiently.

Another study is conducted by [3], where a dataset of 11.682 videos is used to extract insights on online Catch-up TV audiences. One of the key conclusions is the contradiction of the *long-tail* hypothesis, in favor of the *superstar* effect whereby a small fraction of the available programs receive the vast majority of user requests. The conclusions also demonstrate that users tend to request recently aired programs in detriment of older ones. The symbiosis between Catch-up TV and other TV services with users' habits is explored in [33], where a survey is conducted to understand *when*, *how*, and *why* users resort to these services, and how they fit together with their daily routines.

Demand characterization and modeling often motivates and frames design decision in optimization works. An example of this approach is delineated in [20], where a Catch-up TV consumption dataset is used to extract behavioral information and create the so-called Speculative Content Offloading and Recording Engine (SCORE) algorithm, which proactively loads content into users' terminals to reduce peak network bandwidth consumption. In addition to the Catch-up TV characterization, which reinforces and validates the conclusions of other works, the theoretical results show than an oracle with complete knowledge of future requests

is able to achieve remarkable bandwidth and energy savings through an efficient content preloading mechanism.

The work in [13] identifies the need for dynamic network resource provisioning as essential to maintaining a high-QoE in the context of entertainment systems. In this paper, the authors propose the inclusion of a management and control plane responsible for holding a resource prediction engine, combining long and short-term forecasts for resource utilization based on epidemic and time-series models. The forecasts are then used to decided the optimal delivery approach, such as using CDN nodes, or engaging in Peer-to-Peer (P2P) distribution.

The viability of predictive techniques to improve multimedia caching in IPTV environments is explored in [6], where data traces are used to fit synthetic Gaussian, exponential, and power law models which are then used in a modified Least Frequently Used (LFU) caching policy. This technique is shown to outperform the basic Least Recently Used (LRU) algorithm; however, because it assumes historical knowledge on each item being cached, it is not viable in operational environments where new content is added every day.

On a related work, [22], the authors focus on optimizing the delivery of Catch-up TV services, specifically with respect to CDN caching nodes, and propose a content-aware caching algorithm, Most Popularly Used (MPU), which takes advantage of demand forecasts, created using machine learning algorithms, to outclass competing traditional cache replacement policies, such as LRU, LFU, and First-In-First-Out (FIFO).

Even though the issue at hand is focused on multimedia environments, the more general issue of dynamic and autonomic cloud resource management has been explored by other authors. [26] provides an overview on the issues and direction of cloud resource orchestration, which stresses the difficulties associated with dealing with pervasive, highly dynamic and heterogeneous cloud computing resources requiring expert knowledge for deployment, maintenance, monitoring, and control tasks. The need for a resource orchestrator able to forecast and adapt to changes in applications behaviors is identified as a crucial component of the resources' management process.

In [34], a survey is conducted on forecasting and profiling models, which frames the relevance of the problem at hand and systematizes the key motivations behind these techniques, namely: application management; resource management; and cost management. Autonomic resource management is well represented by the MAPE-K (Monitor, Analyze, Plan, Execute, Knowledge) autonomic loop [12], and its related *self-** challenges.

[2] approaches the issue of dynamic resource provisioning in data centers through a reinforcement learning system aiming to reduce job rejection, as its primary goal, while simultaneously minimizing the overall energy consumption, as a secondary and conflicting goal. The results show that the use of machine learning to intelligently manage jobs mostly eliminates job rejections while reducing the total energy consumption.

The existing literature research provides solid foundations and motivations for the present work by showing that content demand forecasting in Catch-up TV is feasible, desirable, are presents a large potential for optimizing several critical CDN aspects, such as caching, bandwidth and capacity planning.

3 Preliminary Data Analysis & Strategy

3.1 Dataset description

The dataset quality is critical for the performance of any forecasting algorithm. In this work, a Catch-up TV consumption dataset is collected from a major IPTV operator and contains 30 days of program request logs, regarding the full month of April 2015, along with Electronic Programming Guide (EPG) metadata

This nonlinear service provides free access to the previous 7 days of program airings on 80 TV channels, depending on users' subscriptions. The content is delivered through a managed network infrastructure using RTSP streams. Each request log entry enables a rich characterization of an individual playback session. Any information that might reveal user details is anonymized. The key dataset information is summarized as follows:

- 22.505.901 requests with device, location, and EPG metadata;
- 704.031 unique households;
- 866.720 unique Set-Top-Boxes;
- 80 unique TV channels;
- Full month of April 2015.

In spite of the large amount of metadata associated with each individual request, not all of it can be used as features (or predictors) for content consumption forecast. In particular, because we are interested in aggregate content consumption, the available dataset is stripped of individual user information, such as location, account and device IDs. Even though the location information may be used to generate consumption forecasts to targeted regions, this information is not essential to the more general Catch-up TV content demand forecasting problem and is left for future work. The key available data fields are:

- *CallLetter*: TV channel identifier. There are 80 distinct TV channels;
- *PlayTime*: Represents the time at which the user requested the program. Given its date/time format, it is not directly usable in prediction models;
- *StartTime*: Initial broadcasting time of the requested programs, as per the EPG. The combination of a unique *StartTime* with a *CallLetter*, unequivocally identifies a program, as each TV channel only airs one program at a time;
- *Duration*: Duration of the requested program in minutes, as per the EPG;
- *IsHD*: Whether a program is available in High-Definition (HD) or Standard Definition (SD). This binary feature is dependent on the channel at which the program was aired, not on the program itself.

Most regression machine learning algorithms require numeric predictors; therefore, the data previously described must be transformed before being suitable for use. The process by which individual, non-usable, features are transformed into predictors applicable on forecasting methods is described as *feature engineering* and is the focus of the next section. When pertinent, the data presented in the figures is normalized so that 100% represents the maximum value, and 0% the minimum value. This normalization maintains the proportionality relationship between the multiple values and does not affect a critical analysis, but avoids disclosing absolute numbers.

3.2 Feature Engineering

The purpose of *feature engineering* is to extract useful numerical predictors out of the dataset, considering the goal of being able to forecast the number of requests that a given program receives within a specific time window.

First, a program must be unequivocally identifiable using one or more predictors. Because a TV channel can only air one program at a time, a natural unique identification is the pair $\{CallLetter, StartTime\}$.

The second input of the forecasting function is the time period to forecast, which may be identified by a continuous time range, with start and end dates, or by discrete ranges such as time slots. The time slot approach fits more naturally with computing systems, by allowing a granularity as flexible as desired, and is employed in this work.

Discrete time slots require a granularity selection able to reach a compromise between accuracy, computing requirements, and usefulness. Our empirical findings and prior data analysis in [23] indicate that 60 minutes time slots represent an adequate compromise between resolution and computing requirements; hence, we define ρ as the time slot unit, with a duration of 3600 seconds (Equation 1).

$$\rho = 3600s \tag{1}$$

Starting with the date features, *PlayTime* and *StartTime*, they are each expanded into additional predictors, describing their day of week and elapsed time of the day. The elapsed time of the day Δt_d of each date d is converted into a discrete time-slot τ_d, according to Equation 2, where S represents the number of time-slots in a day: 24 in the case of 60 minutes slots.

$$\tau_d = \left\lfloor \frac{\Delta t_d}{S} \right\rfloor \tag{2}$$

Having understood the model's inputs, and time-slot transformations, the predictors are expanded as follows:

- *PlayDayOfWeek*: Day of week at which the user request was issued, from 1 (Sunday) to 7 (Saturday);
- *PlaySlotInDay*: Daily time-slot at which the request was issued, ranging from 0 (0:00:00 - 00:59:59) to 23 (23:00:00 - 23:59:59);
- *OriginalDayOfWeek*: Day of week at which the program was originally aired, ranging from 1 (Sunday) to 7 (Saturday);
- *OriginalSlotInDay*: Daily time-slot at which the program was originally aired, ranging from 0 (0:00:00 - 00:59:59) to 23 (23:00:00 - 23:59:59);
- *PlayDelayInSlots*: Time difference, in slots, between the playback request and the program airing;
- *PlayDelayInDays*: Time difference, in days, between the playback request and the program airing.

Fig. 1 provides a graphical overview of each predictor with respect to the total number of requests performed under each one, i.e. the dataset is grouped by each predictor, and their frequency counts are plotted.

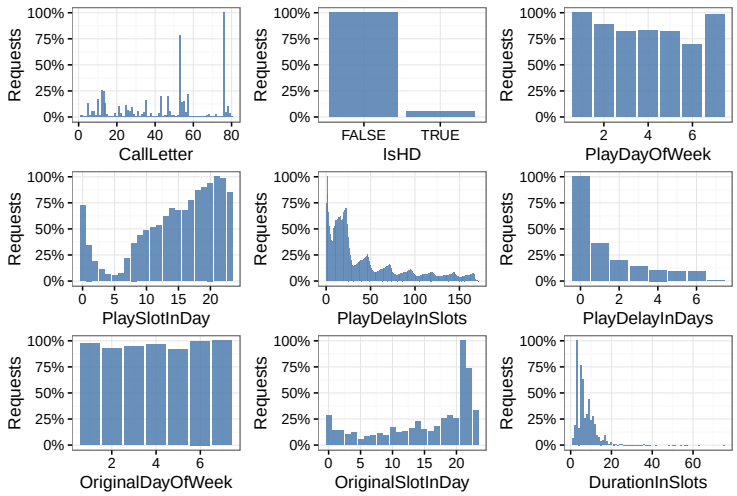


Fig. 1 Initial Feature Engineering Results Plots.

The total number of days available in the dataset is not a multiple of 7; thus, some days of the week are over represented when compared to others (e.g. 5 Thursdays and only 4 Sundays).

Each individual predictor exhibits unique variability patterns with respect to the total number of playback requests; hence, an individual analysis is in order:

- *IsHD*: This predictor clearly shows that SD content is preferred over HD content. The reasons for this behavior may be found in [23];
- *CallLetter*: This is a categorical predictor, with no particular order. It is clear that a few channels represent a disproportionate amount of the total number of playback requests;
- *Duration*: A first observation shows that the playback requests are biased towards content with durations of [20,30], [40,60] and [90,120] minutes;
- *PlayDelayInDays*: There is a clear dominance of requests performed in up to 2 days after the original content aired; thus, the play delay in days appears to be a good fit as a predictor;
- *PlayDelayInSlots*: This predictor reinforces the conclusions of *PlayDelayInDays* with additional detail;
- *OriginalDayOfWeek*: Slight variations in popularity according to the airing day of week;
- *OriginalSlotInDay*: Programs aired on prime-time are more popular;
- *PlayDayOfWeek*: Weekends have a higher number of overall requests, while Friday is the day with the least number of program requests;
- *PlaySlotInDay*: A daily pattern is clear, with a peak of requests at the end of the day, followed by a steep drop in the early morning hours.

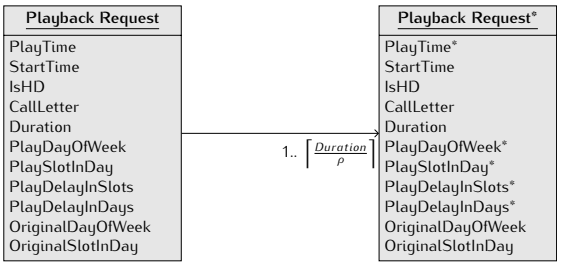


Fig. 2 Playback Requests Mapping into Continuous Sessions.

3.3 Addressing Playback Sessions' Continuity

The dataset only contains records of playback start events, not reflecting the continuous nature of video playback, as a request of a video with 120 minutes of runtime will only be registered once, at the moment of the initial request. This effect is especially important in Catch-up TV scenarios where content is most of the time continuously streamed for its entire duration [20, 23].

The discrete events must, therefore, be able to somehow reflect this continuity; thus, we postulate that each individual event should be replicated to simulate a set of periodic request events up to the total duration of the content. In practice, additional requests are introduced to spread the content over the considered time-slots. Given that we have previously established an acceptable time-slot duration, it makes sense that these emulated events should be created with a period of ρ (Equation 1). A one-to-many mapping is performed, converting each individual playback request into $\left\lceil \frac{Duration}{\rho} \right\rceil$ requests, with adjusted playback time related variables. Fig. 2 illustrates this mapping, which has a significant impact on the data dimensionality. Fields marked with an asterisk ($*$) have been recomputed to take into account the continuity expansion.

Fig. 3 presents the expanded individual feature plots, which are slightly different than those of Fig. 1.

3.4 Forecasting Target

The dataset consists of millions of samples, which, by themselves, are not suitable for building predictive models. On the one hand, no *outcome* variable - the prediction result - has been defined, while on the other hand, the samples are hardly usable due to scalability issues in most machine learning algorithms.

To address these issues, several steps are taken. First, an outcome variable is defined, *RequestsInSlot*, which corresponds to the number of requests that a given Catch-up TV program receives in a specific time-slot of its availability window. Next, for each Catch-up TV program available in the catalog, a matrix is built containing every possible combination of the predictors under consideration: *PlayTime*, *StartTime*, *IsHD*, *Duration*, and so on (refer to Section 3.2 for the complete list of predictors). Then, the outcome variable, *RequestsInSlot*, is added and initialized with 0. This initialization of every possible combination of predictors for each program ensures that there are no missing values, and that the forecasting

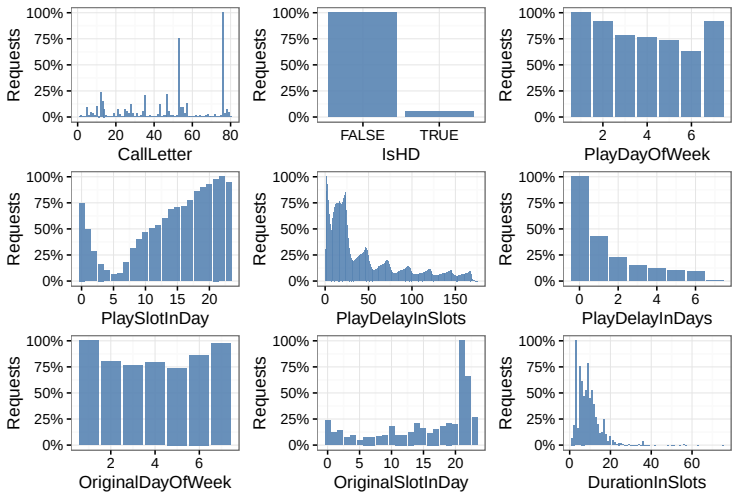


Fig. 3 Expanded Continuous Predictors Results Plots.

Fig. 4 High Level Forecasting Strategy.

models have all the information at their disposal. The next step assigns program requests into each one of these combinations, which in practice amounts to grouping similar entries, counting the number of occurrences, and saving the result in the *ResultsInSlot* variable. In addition to establishing a target variable, this procedure has the advantage of greatly reducing the number of samples to use in model training. With this approach, the forecasting challenge is reduced to predicting the value of *RequestsInSlot* for each combination of predictors.

3.5 Forecasting Strategy

Given the insight on the available features and forecasting target, the issue remains on establishing an adequate strategy for forecasting, especially considering that *IsHD* and *CallLetter* are categorical predictors - that can only take on one of a limited set of values - and the fact that the dataset consists of millions of samples.

A design decision is made to split the forecasting problem per individual channel. This proposition has two main advantages: first, there is no need to convert the *CallLetter* predictor into dummy variables, which would increase the data dimensionality; second, because the prediction is performed for each individual channel, we expect a higher accuracy per channel. This approach does, however, increase the risk of model over-fitting which must be considered.

A high level step-by-step strategy is illustrated in Fig. 4. This process follows industry guidelines for data mining processes, i.e the Cross Industry Standard Process for Data Mining (CRISP-DM) [35].

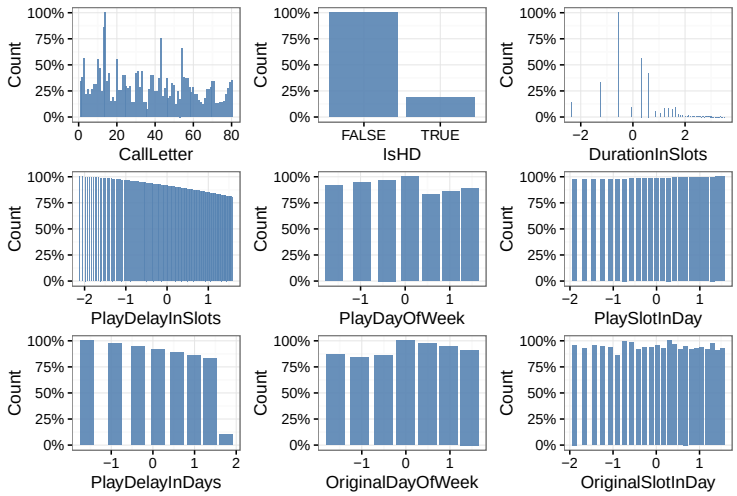


Fig. 5 Feature Plots After Pre-Processing.

4 Pre-Processing & Feature Selection

4.1 Feature Pre-Processing

The different predictors identified in the previous section exhibit different scales, standard deviations, and average values. These discrepancies in scale and statistical properties often impair the numerical stability and bias of learning algorithms, potentially favoring some predictors over others, not because of their real importance but because of their different scales.

In order to compensate for these discrepancies and treat every predictor as equal inputs to learning algorithms, it is important to *scale*, *center*, and correct the *skewness* of each predictor: the *scale* step refers to adjusting the predictors so that their standard deviation is 1; *centering* enforces an average value of 0; and *skewness* correction applies a transformation to the data so that its distribution is more symmetric.

In this work, the pre-processing is performed using the *preProcess* function of the *caret* package [15], whose results are illustrated in Fig. 5 where each individual predictor is corrected for skewness using a Yeo-Johnson transformation [36], centered, and scaled. The Yeo-Johnson's transformation is chosen over the traditional Box-Cox's because it allows for negative and zero values.

As previously mentioned in the forecasting strategy, the categorical predictors *CallLetter* and *IsHD* are not used directly as inputs to the forecasting models; hence, no pre-processing is effectively done on these predictors.

As for the remaining predictors, the changes are noticeable, and show that they are centered at 0 and scaled so that their dimensions are comparable.

4.2 Feature Selection

Feature selection is crucial in properly tuned machine learning models, especially as data dimensionality grows. Having less features to measure or acquire may not only improve the performance of predictive algorithms, as some models are negatively affected by uninformative predictors, but also reduces computational and data acquisition costs. Moreover, models using less predictors are typically more interpretable and better able to adjust to unknown predictors.

Having established that proper feature selection is required for a good performing model, the issue remains on how to select those that improve the models' performance. The two broad classes of feature selection encompass *supervised* and *unsupervised* methods, depending on whether a result or outcome variable is used in the feature selection process (*supervised* or *unsupervised*, respectively).

Both approaches are addressed in the forthcoming subsections, so that their results may be properly compared and a conclusion reached regarding which variables to keep or dismiss.

4.2.1 Unsupervised Feature Selection

This class of feature selection does not depend on the performance of the predictive model to select important features, instead relying on statistical properties to identify the most relevant ones.

One key statistical property is predictor variance. Predictors with zero variance or Near-Zero Variance (NZV), such as those that only have a rare few unique values and a single value for the majority of samples, tend to not carry much information and may generally be discarded.

Predictors with zero variance comprise the edge case that does not add any new information to the model. The general rule of thumb for removing predictors based on NZV is described in [14]:

- Less than 20% of unique values *and*;
- Ratio between the most frequent and the second most frequent must be superior to 20.

Table 1 summarizes the results of the NZV analysis. The *Frequency Ratio* column is defined as the ratio between a predictor's most frequent value and its second most frequent value. The purpose of this ratio is to identify values that are so dominant that do not add any new information, as is usually the case when *Frequency Ratio* > 20. The second column of Table 1, *Percent Unique*, determines the percentage of unique values of a given predictor.

This percentage is mostly useful when combined with the *Frequency Ratio*, given that features with a high *Frequency Ratio* and low *Percent Unique* are good candidates for removal. If both conditions are met, the column NZV evaluates to *true*. The *Zero Variance* column checks if the feature contains a constant value, and is thus redundant.

Another statistical property that is useful for identifying relevant predictors is their cross-correlation. If a dataset contains predictors that are highly correlated ($\rho > 0.95$) there is good chance that these predictors convey the same information; hence, one of them is a good candidate for disposal. The cross correlation plot of

	Frequency Ratio	Percent Unique	Zero Variance	NZV
DurationInSlots	1.766719	0.000415	FALSE	FALSE
PlayDelayInSlots	1.001190	0.001291	FALSE	FALSE
PlayDayOfWeek	1.031823	0.000053	FALSE	FALSE
PlaySlotInDay	1.001284	0.000181	FALSE	FALSE
PlayDelayInDays	1.028900	0.000060	FALSE	FALSE
OriginalDayOfWeek	1.027537	0.000053	FALSE	FALSE
OriginalSlotInDay	1.003541	0.000181	FALSE	FALSE

Table 1 Unsupervised Feature Selection - Variance Analysis.

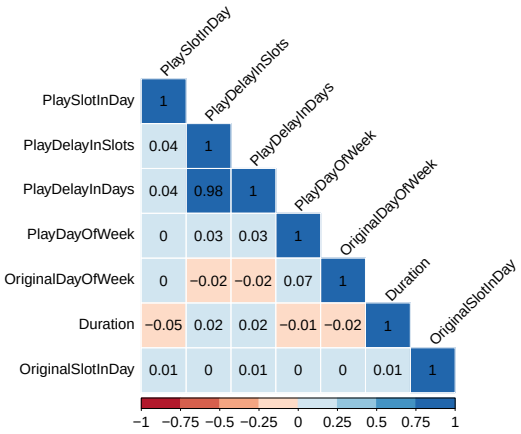


Fig. 6 Unsupervised Feature Selection - Cross-Correlation.

the available predictors is shown in Fig. 6. The most significant observation is that, except for *PlayDelayInDays* and *PlayDelayInSlots*, there is no significant cross-correlation between the different predictors; therefore, they all potentially convey important information.

Considering the results, a choice between predictors *PlayDelayInDays* and *PlayDelayInSlots* must be made. A common heuristic is to keep the predictor that minimizes the average correlation to the remaining predictors [16], which is *PlayDelayInSlots*.

4.2.2 Supervised Feature Selection

Supervised feature selection methods fall into two main classes [9]: *wrapper* and *filter* methods. *Wrapper* methods focus on adding and removing predictors to find the combination that maximizes model performance, and may use genetic algorithms, simulated annealing, recursive feature elimination, and ensemble strategies to name a few. *Filter* methods, on the other hand, conduct evaluations not dependent on the predictive models, and try to find relationships between the predictors and the outcomes in order to select an appropriate set of predictors [30].

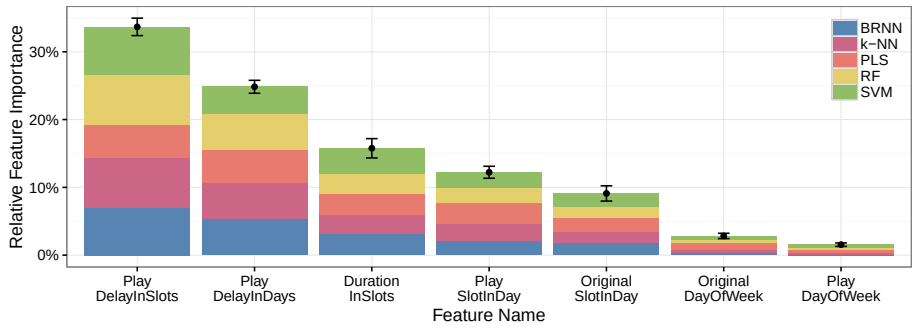


Fig. 7 Ensemble Feature Selection - Weighted Relative Feature Importance

Taking into consideration the possible *filter* and *wrapper* approaches, a wrapper method was selected based on *ensemble* [5] feature selection. This approach was taken due to its known robustness and to the fact that it is easily conducted through the *fscaret R* package [31].

A crucial part of ensemble methods is selecting adequate prediction models from which to obtain the relative importance of each feature. Considering the set of commonly used model types, a subset is chosen that is representative of the main predictive regression models' classes:

- *Bayesian Regularized Neural Networks (BRNNs)*: a class of Neural Networks (NNets) [4, 28];
- *Random Forests (RFs)*: classification and regression based on a forest of trees using random inputs [17];
- *k-Nearest Neighbors (k-NNs)*: widely used in classification and regression [27];
- *Partial Least Squares (PLS)* regression [18];
- *Support Vector Machine (SVM)* with a Radial Basis Function Kernel [10];

Using this set of prediction models, the *fscaret* function is set up to use 10 times repeated 10-fold cross validation, in order to reduce the chance of overfitting the models. The results of Fig. 7 are generated by running the supervised feature selection algorithm using each individual TV channel data, averaging out the final results, and computing the 95% Confidence Interval (CI).

From the results, it is clear that at least two predictors, *OriginalDayOfWeek* and *PlayDayOfWeek*, do not significantly contribute to the overall performance of the predictive model and may be removed.

4.2.3 Final Decision

Considering the results of the unsupervised and supervised feature selection procedures, it is possible to make a decision on the predictors that should be removed before training the regression models.

From the unsupervised feature selection results, the *PlayDelayInDays* is chosen for removal, while from the supervised selection perspective, the *OriginalDayOfWeek* and *PlayDayOfWeek* features should be ignored. A decision is made to

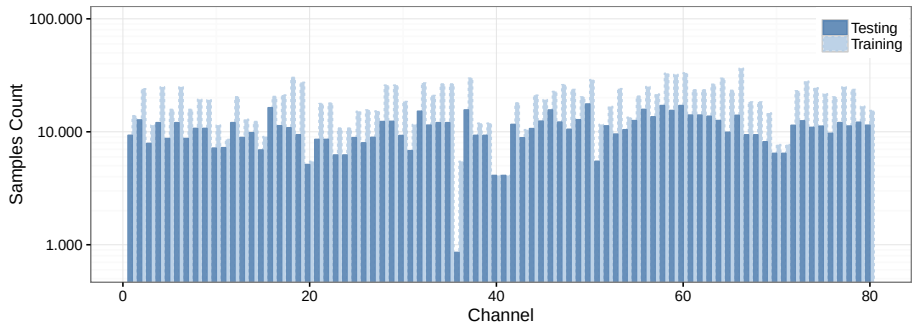


Fig. 8 Forecasting - Available Training and Testing Samples.

take both results into consideration and remove these 3 predictors; thus, the final set of predictors comprises: *PlayDelayInSlots*, *Duration*, *PlaySlotInDay*, and *OriginalSlotInDay*.

These results reinforce the conclusions of Catch-up TV characterization works considered in Section 2, which emphasize the importance of content recency and the time at which programs were originally aired.

5 Model Building Methodology

Before delving into the actual model building and performance testing phase, it is first necessary to establish the tests' conditions and assumptions, along with performance evaluation metrics that ensure the reproducibility, validity, and soundness of the results.

The tests are implemented in *R* [25] using RStudio [29], and run on a Virtual Machine (VM) with 2 Intel E5-2640v3 CPUs (32 cores), and 64GB of RAM.

Even though the performance of the models considered in this analysis are dependent on their actual implementations, the tests are all performed in identical conditions and use reference and commonly used libraries and implementations of the predictive models.

5.1 Training and Testing Procedure

The complete Catch-up TV dataset contains 30 days of user requests logs. A decision is made to split the dataset into two groups. The first comprises the initial 23 days and is used to train the model, while the second relies on the last 7 days and serves to prove that the training process has a good generalization ability in the face of completely unknown data.

In the training phase, 10 different data "folds", or groups, are randomly created from the available samples, and the prediction model is fit using 9 folds. The remaining portion is used for validation and extraction of performance metrics. The whole process is repeated 10 times, with different samples per fold, hence the term 10 times repeated 10-fold cross validation. This particular cross-validation process

is selected because it has been shown to produce similar results as the more computationally burdensome Leave-One-Out Cross-Validation (LOOCV) approach [19].

Fig. 8 provides a graphical overview on the number of training and testing samples available per channel, where it is possible to observe significant disparities between the different channels, both regarding the number of training and testing samples. These disparities are due to the different number of programs aired by each TV channel. As an example, *Kids* channels tend to air many short-duration programs, while *Movies* channels air less programs, but with longer durations [23].

It is also possible to observe that the number of training samples is not $^{23}/_{7}$ times higher than the number of testing samples, as it might be expected. The reason for this apparent mismatch is simple. The forecasting goal, defined in Section 3.4, is to predict the number of requests that a specific Catch-up TV program will receive at each time slot of its availability window, taking into consideration the feature selection procedures performed in Section 4.2. The feature selection process, besides reducing the number of predictors to consider, also creates data samples whose predictors have the exact same value, thus allowing for grouping within each channel's samples. As a result of this grouping procedure, and the higher number of samples that end up being grouped in the training dataset than on the testing dataset — because there are more entries that may be grouped —, the number of samples considered for training and testing are not a multiple of the number of days considered for each group. This process represents a trade-off, as it may reduce the performance of the trained models in exchange for increased training speed.

5.2 Performance Measurements

Even though a common indicator of accuracy is Root Mean Squared Error (RMSE), which is an error measure of the distance between predicted and observed values, the fact that it is scale-dependent [7,32] makes it unfit for comparing the performance of forecasting models for different TV channels, which exhibit very diverse demand volumes, as observed in Fig. 1.

Some of these limitations are addressed by Mean Absolute Scaled Error (MASE) [7,8], in the context of time-series forecasts, as this indicator's scale-free properties enable an accurate comparison of different forecasting algorithms. MASE is composed of two main parts (Equation 3): the numerator computes the average absolute prediction error e_t; the denominator, the *scaling factor*, scales this error with the Mean Absolute Error (MAE) assuming naïve forecasting. Both the numerator and denominator share the original data's scale; hence, MASE is scale-free. In this equation, n is the total number of samples to forecast, while Y_i represents the naïve forecast for period i.

$$\text{MASE} = \frac{\frac{1}{n}\sum_{t=1}^{n}|e_t|}{\frac{1}{n-1}\sum_{i=2}^{n}|Y_i - Y_{i-1}|} \tag{3}$$

Given that MASE was developed to target time-series forecasts, where a natural order exists between the different samples, the scaling factor is easily computed through the naïve forecasting approach. However, in our scenario, no such order exists, and the scaling factor, as defined in MASE, is not appropriate for scaling

the average absolute prediction error. To compensate for this shortcoming, the scaling factor in MASE is replaced by the average outcome of the training set, as per Equation 4, and denominated Training Average Scaled Error (TASE). In the proposed TASE metric, the numerator averages the absolute prediction error e_t for the n total forecasts, while the denominator scales it with the average value of the m training samples. Per the same principle of MASE, TASE exhibits scale-free properties and is used in the evaluations as a key performance indicator.

$$\text{TASE} = \frac{\frac{1}{n}\sum_{t=1}^{n}|e_t|}{\frac{1}{m}\sum_{i=1}^{m}|Y_i|} = \frac{m\sum_{t=1}^{n}|e_t|}{n\sum_{i=1}^{m}|Y_i|} \tag{4}$$

In addition to evaluating the models' fitness, other metrics of practical nature are also considered, especially with regards to the computational time required to train and test each model as this is a limiting factor on the practical deployment of a forecasting solution.

5.3 Data Selection Procedure

Due to computational limitations, some tests do not use the full dataset at our disposal; therefore, an adequate data selection method is required.

The limitations in question are due to the trade-off between the number of CPUs and RAM memory usage, which requires a careful balance. Using more CPUs leads to faster model training, but also to a proportional increase in RAM memory usage, which must be kept within the total memory budget. These limitations are particularly relevant on the grid search performed in Section 6.1, which require training forecasting models with several algorithms and parameters.

The data selection aspects are handled by the *createDataPartition* function of the *caret* package [15], which uses statistical information to group data into percentiles and subgroups, which are then randomly sampled. As our target is to produce demand forecasts for each time-slot within a given day, this function is used to sample data from each channel using the *PlaySlotInDay* predictor.

6 Results and Discussion

Having performed a proper feature selection and established a test methodology, this final step deals with creating and selecting a forecasting model, based on a cost-performance trade-off analysis.

The choice of a forecast regression method must take into consideration not only the algorithm's forecasting accuracy, but also its computational demand, and its ability to be rebuilt or updated as new data becomes available.

As the performance metrics chosen are scale-free, the individual channels' results are directly comparable. Whenever pertinent, the 95% confidence interval is shown as a shaded area surrounding the average value curve and data points.

6.1 Tuning Parameters

Before proceeding with performance testing, it is first necessary to find adequate tuning parameters for each model. Properly tuned models are essential to produce

good results. Each model type has its own set of tune parameters that must be adequately configured to maximize their performance.

In order to determine suitable parameters for each predictive algorithm, a grid search is conducted. The grid search is an hyperparameter optimization approach whereby an exhaustive search is conducted using a set of manually specified parameters to determine which parameter combination yields the best model performance, according to the previously defined TASE metric. These tuning parameters — or variables — are different per model, and must obey to distinct constraints. To reduce the chance of over-fitting, cross-validation is performed according to Section 5.1.

A maximum of 10.000 samples are selected from each channel's training data, according to the selection procedures defined in Section 5.3. The number of samples selected is the result of empirical evaluations where it was found that the time required to train each algorithm for larger sample sizes, and for each combination of parameters took more than 1 day in some cases, without producing noticeably better results.

6.1.1 Bayesian Regularized Neural Network (BRNN)

BRNN's [4, 28] implementation is based on a two layer neural network. Its key parameters, and default values, enclosed in parentheses, are as follows:

- *neurons* (2): in the hidden layer;
- *epochs* (1000): the maximum iterations to train;
- μ (0.005): Marquardt adjustment parameter;
- μ_{dec} (0.01): Decrease factor for μ;
- μ_{inc} (10): Increase factor for μ;
- μ_{max} ($1 \cdot 10^{10}$): Maximum value for μ;

Isolated tests showed that the key adjustment parameter is the number of *neurons*; therefore, to prevent a very extensive grid search, the parameter tuning process for BRNNs is focused on the number of *neurons*, and the remaining configurations are set with their default values. While no optimal number of *neurons* is provided by the existing literature, rules of thumb exist that are used to limit the grid search, from which a common one [11] is to use twice the number of inputs, leading to a total of 8 neurons. To account for the rule of thumb's imprecisions, the set $\{2, 3, 4, 5, 6, 7, 8, 9, 10\}$ is used for total number of *neurons*.

6.1.2 k-Nearest Neighbor (k-NN)

k-NN's [27] single tune parameter is the number of neighbors (k) considered. A too-low value of k may lead to under-fitting, while a high k may cause over-fitting. A decision is made to use the set: $\{5, 7, 9, 11, 13, 15, 17, 19\}$ of k-neighbors.

This k-NN implementation relies on the Euclidean metric to determine the neighbors' distance.

6.1.3 Partial Least Squares (PLS)

PLS [18] has a single tuning parameter, the number of components (*ncomp*), which is limited to the number of features used to train the model; therefore, the search set is: $\{1, 2, 3, 4\}$.

6.1.4 Random Forest (RF)

The RF [17] implementation used in the benchmarks has the following tuning parameters:

- *mtry* (1): number of variables randomly sampled as candidates at each split. Its maximum is limited by the total number of features;
- *ntree* (500): number of trees to grow ;
- *nodesize* (5): corresponds to the minimum size of terminal nodes and controls the tree depth. A larger *nodesize* lead to smaller trees.

Given that *ntree* is large enough to reasonably ensure that all the available features (4) will be evaluated at least once, and that the default *nodesize* of 5 leads to deep trees — due to the large sample size —, their default values are adequate. Therefore, the tune parameter to optimize in RFs is *mtry* with the search set of $\{1, 2, 3, 4\}$.

6.1.5 Support Vector Machine (SVM)

The SVM [10] model in use relies on a radial basis kernel (Gaussian), and has the following tuning parameters:

- C (1): cost of constraints violation, which penalizes excessive slacks;
- *sigma*: used to map inputs into a feature space;

To limit the grid search computational demand, a choice is made to compute *sigma* according to literature recommendations, using *sigest* of the *kernlab* package [10], and to vary C instead. Using empirical evaluations, C is varied using powers of 2 according to the set $\{2, 4, 8, 16, 32, 64, 128, 256, 512\}$.

6.1.6 Tuning Results

Fig. 9 presents the grid search results by individual model, from which several conclusions may be drawn.

Starting with BRNNs, where the tuning parameter is the number of *neurons*, it is clear that its performance improves with the addition of up to 8 neurons, after which the models' performance stabilizes.

As for k-NN, the results show that the cost of generalization, translated into a higher number of neighbors (k), is a decline in global performance; hence, k is set to 5 for the forthcoming performance evaluations. PLS models have a completely different behavior and are shown to perform roughly the same regardless of the chosen number of components. The adequate number of components, *ncomp*, selected for the model training phase is 3.

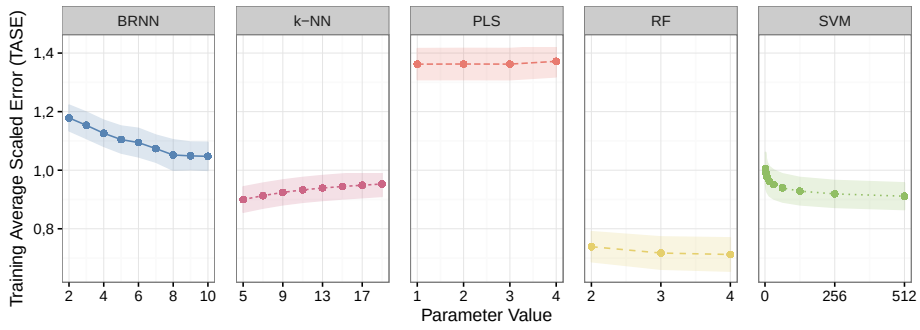


Fig. 9 Tuning Parameter Selection.

In this tuning phase, RFs show the best performance of the considered models, especially when using the maximum number of *mtry*, 4.

Finally, SVMs' performance improves with an increase in cost, which stabilizes for metric costs over 128. The best value of 512 is chosen for C.

6.2 Statistical Model Evaluation

The issue of scaling is critical on most systems, and often makes or breaks the applicability of a model or algorithm. Taking into consideration that the regressive forecasting models evaluated in this study are fundamentally very different, it is important to understand how they scale with the training set size, specifically in terms of computational requirements and accuracy.

To explore this compromise, a parametric analysis is conducted by varying the training sets sizes, per individual TV channel, and observing the corresponding changes in TASE, R^2, training and forecasting time. This procedure is performed according to the following specifications:

- 80 TV Channels;
- 23 first days used for training, with 10 times repeated 10-fold cross validation, and 7 days used for additional performance testing;
- 10 times repeated 10-fold cross validation;
- The maximum number of training samples per channel is varied according to the set: {1000, 2000, 5000, 7000, 10000, 15000, 20000, 30000};
- Performance metrics under consideration:
  - *Training Average Scaled Error (TASE)*;
  - R^2: squared Pearson's correlation coefficient;
  - *Training time*: time required to build the model;
  - *Forecasting time*: time taken to produce the model's predictions.

6.2.1 Training Time

One of the most important scaling issues on predictive models is their training time, which provides a hard constraint on which models are usable or not.

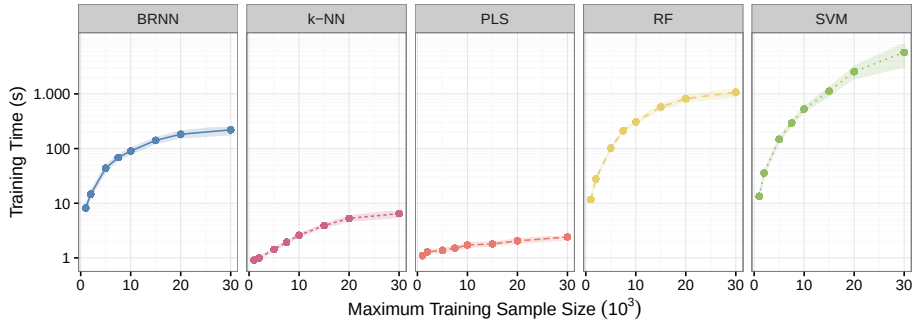


Fig. 10 Training Time Scaling with Training Sample Size.

Fig. 10 presents the average training time for the predictive models considered, where it is possible to observe very large discrepancies between them. This training time is collected on a per-channel basis.

It is clear from the results that SVM provides the worst scaling behavior, which may limit it to training models with less than 20.000 samples. As for the remaining models, k-NN and PLS exhibit a conservative training time growth, while BRNN and RF scale better than SVM.

The training time slope decay on all models after approximately 20.000 samples is due to the channels that do not have more than these samples available, as discussed in Section 5 and seen in Fig. 8. While the training times for maximum sample sizes greater than 20.000 are affected by this limitation, the results are still comparable between the different forecasting models.

6.2.2 Forecasting Time

The forecasting time metric is also important when selecting a predictive model, as the time it takes to produce results, and how it grows with the number of predicted values, has an impact in the design of systems which expect quick responses from predictive models to make a decision, especially if the system has to operate in a real-time, or quasi-real-time scenario.

The results presented in Fig. 11 show that the models have different behaviors with respect to the required forecasting time, and scale differently than on the previous analysis. From the evaluated models, PLS is the fastest to provide an outcome estimate, closely followed by BRNN. RF and k-NN fare worst, but better than SVM, which displays the worst performance.

Regardless of the forecasting models, the time required to produce forecasts is always reasonable, and significantly lower than the required training times.

6.2.3 TASE

This scaling analysis focuses on TASE, described in Section 5.2, and is presented in Fig. 12. As previously mentioned, TASE provides a scale-free error metric that is suitable for comparing the performance of the forecasting models between different TV channels, which exhibit distinct demand profiles and scales. The lower the

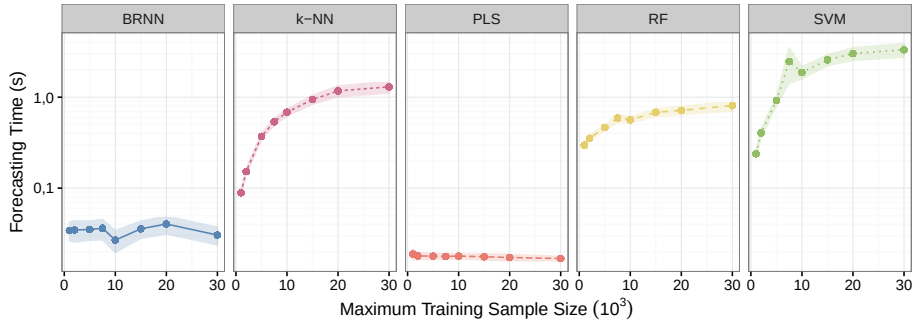


Fig. 11 Forecasting Time Scaling with Training Sample Size.

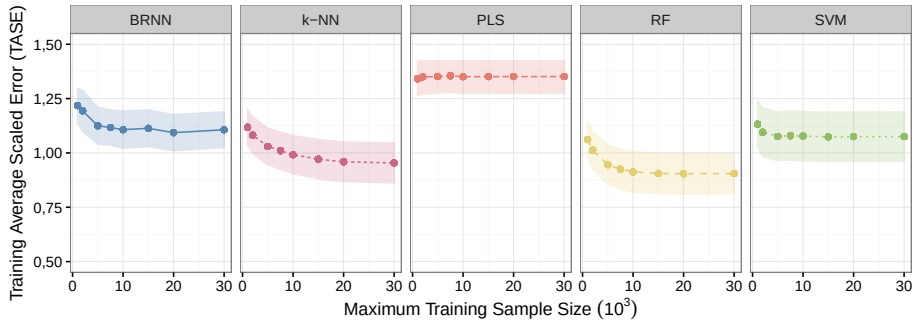


Fig. 12 TASE Scaling with Training Sample Size.

TASE, the better the prediction, with 0 corresponding to a perfect forecast, i.e. the predictions match the observations.

The first observation is that the models' performance with respect to TASE does not appear to vary significantly with training set sizes greater than 10.000 samples.

When considering each model individually, additional conclusions may be withdraw. PLS provides the worst results, which is expected due to the model's simplicity, when compared with alternative approaches. BRNN fares significantly better than PLS, but worse than the other competing models. SVM's performance appears to be somewhat insensitive to maximum training sample sizes over 2.000, which is remarkable as it fares better than BRNN. The clear winner is RF, whose performance improves greatly from small sample sizes up to 10.000 maximum training samples, after which the performance gains are reduced. Lastly, k-NN provides a middle-ground performance between RF and SVM, especially for higher maximum training sample sizes.

6.2.4 R^2

Evaluating the correlation between the predicted and observed outcomes provides an insight on a model's capability of explaining the actual data variation, instead

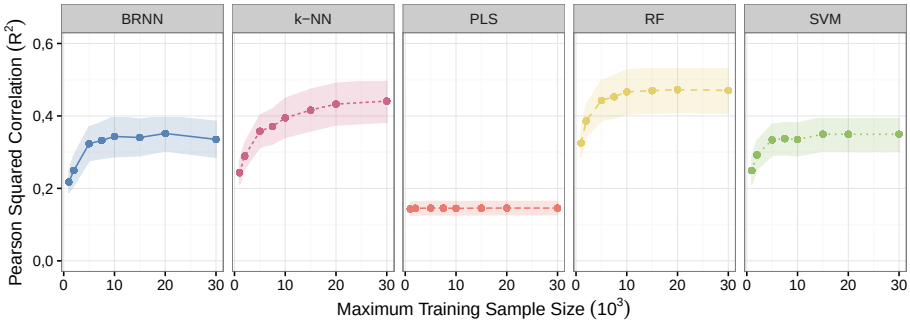


Fig. 13 R^2 Scaling with Training Sample Size.

of how accurate the prediction is. An ideal model would provide a correlation metric R^2 of 1.

PLS displays the worst performance in terms of R^2, which is approximately constant regardless of the training sampling size. The remaining models, BRNN, k-NN, RF, and SVM, exhibit a wide confidence interval with respect to R^2, which indicates a large variability among the models trained for each TV channel.

In the case of BRNN, k-NN, RF, and SVM, R^2 improves with the maximum training sampling size, but its performance approximately stabilizes when models are trained with a maximum of 20.000 samples.

The overall results appear similar to that of TASE scaling evaluation, with BRNN and SVM producing results that are better than PLS but worse than k-NN and RF. RF provides the best overall performance in terms of R^2.

6.2.5 Comparative Evaluation

Taking into consideration the results obtained from training these 5 representative predictive models, it is possible to make a decision on the most appropriate one for forecasting Catch-up TV consumption.

Starting with PLS, in spite of its low training and forecasting times, along with its scalability with the training set size, the disappointing TASE and R^2 results show that it fares much worse than other competing models.

SVM and BRNN do not stand-out with respect to their forecasting performance. SVM requires a very large training time to achieve results similar to BRNN, while also requiring the largest forecasting time to predict the demand of new samples.

The best results come from k-NN and RF models, whose performance increases with the number of maximum training samples, and show good TASE and R^2 results. From the two, RF provides the best results, at the expense of an increased training time when compared to k-NN.

Considering the performance trade-offs involved, particularly with respect to the training and forecasting times required, the sweet-spot for the maximum number of training samples appears to be 20.000, as the gains of using a higher number of samples in terms of TASE and R^2 are residual, especially for best performing predictive models, k-NN and RF.

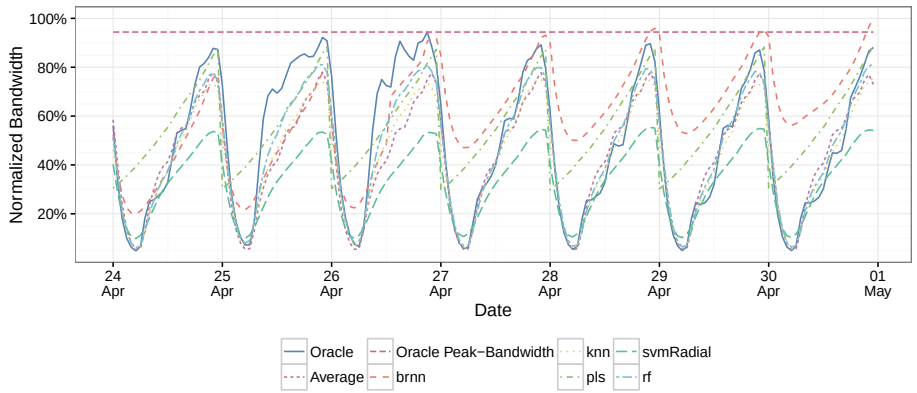


Fig. 14 Bandwidth Requirements Forecast.

6.3 Dynamic Resource Provisioning

To gauge the benefits achievable by demand forecasting mechanisms in OTT delivery systems, an analysis is performed on bandwidth and storage requirements and how they vary throughout the testing period.

Results are presented in a normalized fashion, ranging from 0% to 100%, to facilitate a graphical analysis. According to the previously described methodology, forecasts are produced for the testing period, i.e. last 7 days of April 2015, after using the initial 23 days for training.

In addition to depicting results from the considered machine learning algorithms, two baselines are added for comparison purposes. The first, *Oracle*, represents the actual observed values throughout the 7 days forecasting period, while the second one, *Average*, provides a static analysis of past performance and represents the average values observed in the training period.

6.3.1 Bandwidth

The issue of bandwidth requirements is prevalent in computer networks, especially in those dedicated to bandwidth-intensive multimedia streaming. To understand how they vary with time, Fig. 14 provides a comparison of the *Oracle*, *Average*, and predicted bandwidth demand per forecasting model. An additional helper curve, *Oracle Peak-Bandwidth*, is added to represent the maximum observed bandwidth during the forecasting period.

These results fall in line with the previously performed statistical evaluation, and showcase the models' demand forecasting abilities. RF and k-NN exhibit the best overall performance, closely tracking the observed bandwidth requirements and improving over the *Average* demand baseline. BRNN and SVM provide a lesser approximation of the bandwidth demand curve, with BRNN sometimes over-predicting demand for the late night hours, and SVM consistently under-predicting peak-demand. As expected, PLS provides the worst approximation. Overall, except for PLS, all forecasting models are able to provide a demand forecast that approximates the observed demand.

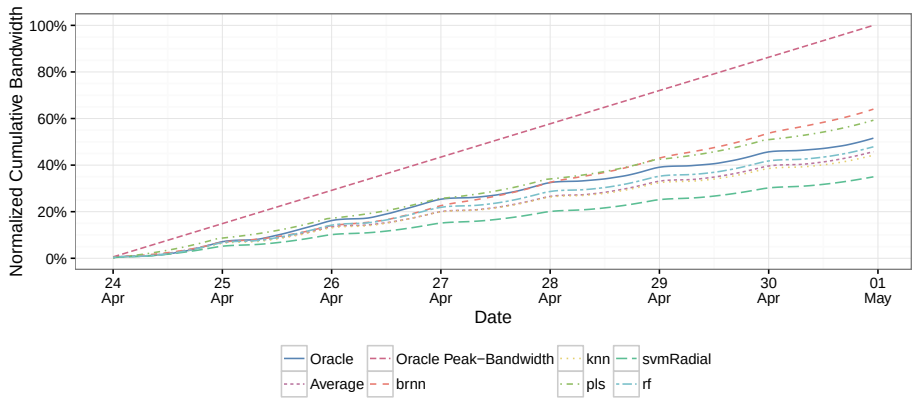


Fig. 15 Bandwidth Savings.

To complement these demand forecast results, Fig. 15 provides an analysis on the cumulative bandwidth requirements. This point-of-view allows a better insight on the potential power and cost savings.

Oracle Peak-Bandwidth provides an upper bandwidth limit and corresponds to static provisioning at maximum capacity; *Oracle* presents the actual bandwidth demand; *Average* represents the average bandwidth demand according to historical data; finally, the prediction curves per machine learning model are presented.

The results indicate that less than 50% of the total statically provisioned resources, *Oracle Peak-Bandwidth*, are required to address the dynamic demand; therefore, an ideal resource provisioning system, with a linear relationship between power consumption and cost, would be able to use less than 50% of the total power, and reduce more than 50% of costs. In practice, the actual relationship between provisioned resources, cost, and power consumption is not this simple, but these results provide a ball-park indicator of the potential savings.

6.3.2 Storage

Other potential application of demand forecasting systems for Catch-up TV is storage optimization. Previous studies have shown that users do not take advantage of the complete content catalog at their disposal [23], leading to unnecessary storage provisions.

To address this issue, an analysis is conducted with the purpose of assessing, at each time of day, the programs actually requested by users and their storage requirements. These requirements are determined as a function of the content's duration and video quality. For the considered Catch-up TV service, HD content requires twice the storage amount per unit of time than the SD counterpart.

Fig. 16 conveys the results of this investigation and shows that, similarly to the bandwidth analysis, very significant gains can be achieved by taking into consideration demand forecasts.

The *Available* curve reflects the total storage required to hold the complete Catch-up TV content catalog; the *Oracle* curve shows the storage requirements of the actually requested content; the *Average Storage* curve is presented to reflect

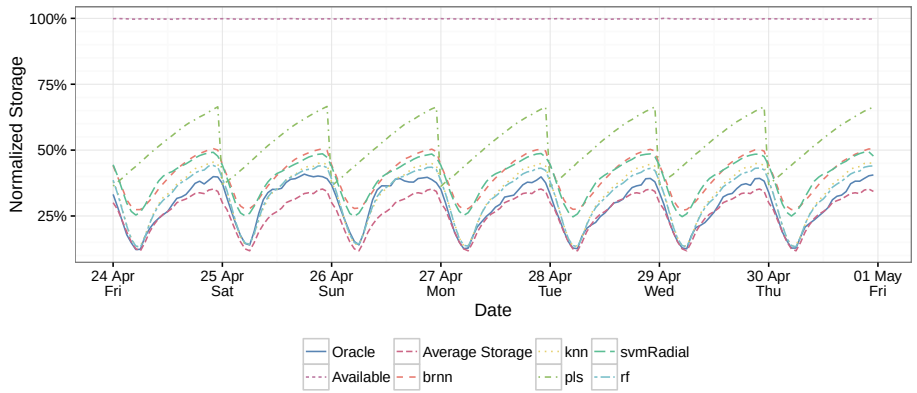


Fig. 16 Storage Savings.

the static analysis over historical data; lastly, individual curves are shown per machine learning model.

The slight variations in the *Available* storage requirements curve is due to Catch-up TV content being added and removed from the content catalog throughout the day.

The forecast results vary according to the underlying machine learning model used, with RF providing the most accurate results, and PLS the worst. All machine learning models slightly overestimate the actual storage requirements. In spite of the over-estimation, it is possible to observe that the storage requirements correspond to a fraction of the content catalog, peaking at under 50% of the total *Available* catalog.

6.3.3 Summary

Considering the results, it is possible to observe that the predictive models are accurate enough to produce usable bandwidth and storage requirements forecasts to be used in dynamic operational environments. RF and k-NN produced the most accurate predictions and surpass the performance of static historical analysis.

Significant bandwidth and storage savings are possible in dynamic provisioning environments, leading to potentially large savings and cost and power consumption. The results provide an indication that the service performance, on average, should not be affected, as it ensures that there enough available resources to meet the demand. However, the actual service performance, from a QoE perspective, is very dependent on the statistical characterization of user demand, particularly in terms of its variance during the forecasting time-slot. To compensate for demand fluctuations a slight over-provisioning may be required.

The chosen time-slot duration of 1 hour proves to be adequate for generating accurate forecasts. Shorter forecasting time-slots are possible, but are constrained by computational requirements, in spite of their potential for even more dynamic adjustments.

7 Conclusion & Future Work

Forecasting content demand is a formidable challenge with applications on several scientific and industrial areas. This study shows how this challenge may be addressed in the context of Catch-up TV delivery optimization.

A step-by-step approach to building a predictive model able to leverage historical consumption data to produce accurate estimations is provided. After detailing the feature pre-processing and engineering process, the most suitable features are selected by combining supervised and unsupervised methods.

The performance results compare 5 reference machine learning algorithms and show that RFs are able to outperform BRNN, k-NN, SVM, and PLS, while requiring reasonable training and forecasting times.

The dynamic resource provisioning study shows that the forecasting models are able to produce accurate bandwidth and storage requirements forecasts, which may be used to achieve considerable power and cost savings.

These promising results provide a starting point for future work on dynamic and adaptive OTT CDNs capable of delivery next-generation multimedia content in an efficient and cost effective manner whilst maintaining a high QoE.

Future work will include a study to determine and characterize the impact of prediction-based resource provisioning on QoE to assess if, and how much, over-provisioning is required to ensure that no negative reflection on service performance exists.

Acknowledgements The authors would like to thank Fausto Carvalho (Altice Labs, SA) and João Ferreira (MEO - Serviços de Comunicações e Multimédia, SA) for the key discussions and for providing the raw Catch-up TV dataset.

This research was funded by UltraTV (Portugal 2020 POCI-01-0247-FEDER-017738), by FCT/MEC through national funds, and when applicable co-funded by FEDER PT2020 partnership agreement under the project UID/EEA/50008/2013 OT2Delivery (Over-the-top Multimedia Content Delivery for Next Generation Mobile Networks).

References

1. ANACOM: Subscription Television Service Statistical Information 2nd Quarter 2015. Tech. rep., ANACOM (2015). URL `http://www.anacom.pt/streaming/STVS2quarter2015.pdf?contentId=1366508&field=ATTACHED_FILE`. Accessed: 12-2015
2. Bahrpeyma, F., Haghighi, H., Zakerolhosseini, A.: An adaptive RL based approach for dynamic resource provisioning in Cloud virtualized data centers. Computing **97**(12), 1209–1234 (2015). DOI 10.1007/s00607-015-0455-8. URL `http://dx.doi.org/10.1007/s00607-015-0455-8`
3. Beauvisage, T., Beuscart, J.S.: Audience dynamics of online catch up TV. In: Proceedings of the 21st international conference companion on World Wide Web - WWW '12 Companion, p. 461. ACM Press, New York, USA (2012). DOI 10.1145/2187980.2188077. URL `http://dx.doi.org/10.1145/2187980.2188077`
4. Burden, F., Winkler, D.: Bayesian Regularization of Neural Networks. In: D.J. Livingstone (ed.) Artificial Neural Networks, *Methods in Molecular Biology*, vol. 458, pp. 23–42. Humana Press (2009). DOI 10.1007/978-1-60327-101-1_3. URL `http://dx.doi.org/10.1007/978-1-60327-101-1_3`
5. Caruana, R., Niculescu-Mizil, A., Crew, G., Ksikes, A.: Ensemble selection from libraries of models. In: Twenty-first international conference on Machine learning - ICML '04, vol. 34, p. 18. ACM Press, New York, New York, USA (2004). DOI 10.1145/1015330.1015432. URL `http://dx.doi.org/10.1145/1015330.1015432`

6. Famaey, J., Iterbeke, F., Wauters, T., De Turck, F.: Towards a predictive cache replacement strategy for multimedia content. Journal of Network and Computer Applications **36**(1), 219–227 (2013). DOI 10.1016/j.jnca.2012.08.014. URL http://dx.doi.org/10.1016/j.jnca.2012.08.014
7. Hyndman, R.: Another Look At Forecast-Accuracy Metrics for Intermittent Demand. Foresight: The International Journal of Applied Forecasting (4), 43–46 (2006). URL http://www.researchgate.net/publication/5055536_Another_Look_at_Forecast_Accuracy_Metrics_for_Intermittent_Demand/file/d912f50ff0c2ad9136.pdf. Accessed: 01-2016
8. Hyndman, R.J.: forecast: Forecasting Functions for Time Series and Linear Models (2015). URL https://cran.r-project.org/web/packages/forecast/index.html. Accessed: 01-2016
9. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant Features and the Subset Selection Problem. In: W.W. Cohen, H. Hirsh (eds.) Machine Learning: Proceedings of the Eleventh International Conference, pp. 121–129. Morgan Kaufmann, San Francisco, CA (1994). URL http://machine-learning.martinsewell.com/feature-selection/JohnKohaviPfleger1994.pdf. Accessed: 09-2015
10. Karatzoglou, A., Smola, A., Hornik, K.: kernlab (2015). URL https://cran.r-project.org/web/packages/kernlab/kernlab.pdf. Accessed: 09-2015
11. Karsoliya, S.: Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture. International Journal of Engineering Trends and Technology (IJETT) **3**(6), 714–717 (2012). URL http://www.ijettjournal.com/volume-3/issue-6/IJETT-V3I6P206.pdf
12. Kephart, J., Chess, D.: The vision of autonomic computing. Computer **36**(1), 41–50 (2003). DOI 10.1109/MC.2003.1160055. URL http://dx.doi.org/10.1109/MC.2003.1160055
13. Kryftis, Y., Mastorakis, G., Mavromoustakis, C.X., Batalla, J.M., Pallis, E., Kormentzas, G.: Efficient entertainment services provision over a novel network architecture. IEEE Wireless Communications **23**(1), 14–21 (2016). DOI 10.1109/MWC.2016.7422401. URL http://dx.doi.org/10.1109/MWC.2016.7422401
14. Kuhn, M.: Building Predictive Models in R Using the caret Package. Journal of Statistical Software **28**(5), 1–26 (2008). URL http://www.jstatsoft.org/v28/i05. Accessed: 09-2015
15. Kuhn, M.: caret (2015). URL https://cran.r-project.org/web/packages/caret/caret.pdf. Accessed: 09-2015
16. Kuhn, M., Johnson, K.: Applied Predictive Modeling. Springer New York, New York, NY (2013). DOI 10.1007/978-1-4614-6849-3. URL http://dx.doi.org/10.1007/978-1-4614-6849-3
17. Liaw, A.: randomForest (2015). URL https://cran.r-project.org/web/packages/randomForest/randomForest.pdf. Accessed: 09-2015
18. Mevik, B.H., Wehrens, R., Liland, K.H.: pls (2015). URL https://cran.r-project.org/web/packages/pls/pls.pdf. Accessed: 09-2015
19. Molinaro, A.M., Simon, R., Pfeiffer, R.M.: Prediction error estimation: a comparison of resampling methods. Bioinformatics **21**(15), 3301–3307 (2005). DOI 10.1093/bioinformatics/bti499. URL http://bioinformatics.oxfordjournals.org/cgi/doi/10.1093/bioinformatics/bti499
20. Nencioni, G., Sastry, N., Chandaria, J., Crowcroft, J.: Understanding and decreasing the network footprint of catch-up tv. In: Proceedings of the 22nd international conference on World Wide Web - WWW '13, pp. 965–976. ACM Press, New York, New York, USA (2013). DOI 10.1145/2488388.2488472. URL http://dx.doi.org/10.1145/2488388.2488472
21. Nielsen: The Digital Consumer (2014). URL http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2014Reports/the-digital-consumer-report-feb-2014.pdf. Accessed: 09-2015
22. Nogueira, J., Gonzalez, D., Guardalben, L., Sargento, S.: Over-The-Top Catch-up TV Content-Aware Caching. In: 21st IEEE Symposium on Computers and Communication (ISCC), p. 6. Messina, Italy (2016)
23. Nogueira, J., Guardalben, L., Cardoso, B., Sargento, S.: Catch-up TV analytics: statistical characterization and consumption patterns identification on a production service. Multimedia Systems -, 1–19 (2016). DOI 10.1007/s00530-016-0516-7. URL http://dx.doi.org/10.1007/s00530-016-0516-7

24. Pathan, M., Buyya, R.: A Taxonomy of CDNs. In: Content Delivery Networks, chap. A Taxonomy, pp. 33–77. Springer Berlin Heidelberg, Berlin, Heidelberg (2008). DOI 10.1007/978-3-540-77887-5_2. URL http://dx.doi.org/10.1007/978-3-540-77887-5_2
25. R Foundation for Statistical Computing: The R Project for Statistical Computing (2016). URL https://www.r-project.org/. Accessed: 01-2016
26. Ranjan, R., Benatallah, B., Dustdar, S., Papazoglou, M.P.: Cloud Resource Orchestration Programming: Overview, Issues, and Directions. IEEE Internet Computing **19**(5), 46–56 (2015). DOI 10.1109/MIC.2015.20. URL http://dx.doi.org/10.1109/MIC.2015.20
27. Ripley, B., Venables, W.: class (2015). URL https://cran.r-project.org/web/packages/class/class.pdf. Accessed: 09-2015
28. Rodriguez, P.P., Gianola, D.: brnn (Bayesian regularization for feed-forward neural networks) (2015). URL https://cran.r-project.org/web/packages/brnn/brnn.pdf. Accessed: 01-2016
29. RStudio Inc.: RStudio (2016). URL https://www.rstudio.com/. Accessed: 01-2016
30. Saeys, Y., Inza, I.n., Larranaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics **23**(19), 2507–2517 (2007). DOI 10.1093/bioinformatics/btm344. URL http://dx.doi.org/10.1093/bioinformatics/btm344
31. Szlek, J., Mendyk, A.: fscaret (2015). URL https://cran.r-project.org/web/packages/fscaret/fscaret.pdf. Accessed: 09-2015
32. Tofallis, C.: A better measure of relative prediction accuracy for model selection and model estimation. Journal of the Operational Research Society **66**(8), 1352–1362 (2015). DOI 10.1057/jors.2014.103. URL http://dx.doi.org/10.1057/jors.2014.103. Accessed: 01-2016
33. Vanattenhoven, J., Geerts, D.: Broadcast, Video-on-Demand, and Other Ways to Watch Television Content. In: Proceedings of the ACM International Conference on Interactive Experiences for TV and Online Video - TVX '15, pp. 73–82. ACM Press, New York, New York, USA (2015). DOI 10.1145/2745197.2745208. URL http://dx.doi.org/10.1145/2745197.2745208
34. Weingärtner, R., Bräscher, G.B., Westphall, C.B.: Cloud resource management: A survey on forecasting and profiling models. Journal of Network and Computer Applications **47**, 99–106 (2015). DOI 10.1016/j.jnca.2014.09.018. URL http://dx.doi.org/10.1016/j.jnca.2014.09.018
35. Wirth, R.: CRISP-DM : Towards a Standard Process Model for Data Mining. Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining **-**, 29–39 (2000). DOI 10.1.1.198.5133. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.198.5133. Accessed: 12-2015
36. Yeo, I.K., Johnson, R.A.: A new family of power transformations to improve normality or symmetry. Biometrika **87**(4), 954–959 (2000). DOI 10.1093/biomet/87.4.954. URL http://dx.doi.org/10.1093/biomet/87.4.954

Appendix F

Over-The-Top Catch-up TV Content-Aware Caching.

Over-The-Top Catch-up TV Content-Aware Caching

João Nogueira*†, Daniel Gonzalez†‡§, Lucas Guardalben†‡ and Susana Sargento†‡
*Altice Labs, SA, Aveiro, Portugal
†University of Aveiro, Aveiro, Portugal
‡Instituto de Telecomunicações, Aveiro, Portugal
§Carlos III University of Madrid, Madrid, Spain
joaonogueira@ua.pt, dmartin@ua.pt, guardalben@ua.pt, susana@ua.pt

***Abstract*—The migration of popular Catch-up TV services to modern Over-The-Top (OTT) multimedia delivery infrastructures creates a wide set of scalability challenges which are commonly addressed using Content Delivery Networks (CDNs) relying on caching nodes close to users.**

The use of general-purpose caching nodes, tailored for generic web content, is far from optimal as it does not consider the particularities of Catch-up TV content, namely its dynamic popularity behavior, superstar effects, and relevance decay, as shown in existing scientific literature. Since caches are limited in size and are relatively small when compared to the whole catalog of available Catch-up TV content, which may contain tens of thousands of TV programs, it is crucial to make the most out of the available resources.

To address these issues, this paper proposes a novel content-aware cache replacement algorithm, Most Popularly Used (MPU), capable of taking advantage of content demand forecasts built using machine learning models, to significantly outperform traditional cache replacement policies, such as Least Recently Used (LRU), Least Frequently Used (LFU), and First-In-First-Out (FIFO), and approach the optimal theoretical hit-ratio limits. MPU leverages millions of Catch-up TV request logs to validate its results under realistic conditions.

1. Introduction

Large scale delivery of Catch-up TV content is a challenge faced by IP Television (IPTV) operators struggling to cope with the service's growing demand, whose popularity is orders of magnitude larger than that of traditional Video-on-Demand (VoD) content [1], [2].

This struggle is particularly relevant for operators transitioning from managed to unmanaged delivery, i.e. to Over-The-Top (OTT), with the purpose of achieving the *anytime-anywhere* promise of convergent solutions while simultaneously lowering the overall Capital Expenditures (CAPEX) and Operational Expenditures (OPEX) requirements of this resource-intensive multimedia service [3], [4], [5].

In order to address the scalability challenges in OTT multimedia delivery scenarios, a common approach is to employ CDNs to get the content close to its users, minimize bandwidth costs, and improve users' Quality-of-Experience (QoE); however, the use of CDNs is rife with difficulties, ranging from caching optimization to ensuring adequate bandwidth allocations, and selecting Point-of-Presence (PoP) locations, to name a few.

Given that the overall performance of CDNs is highly dependent on the efficiency of caching nodes, measured in hit-ratios and upstream bandwidth savings, and that the modification of caching algorithms is a feasible operation in commonly used proxy cache solutions, such as Apache Traffic Server (ATS) [6], Nginx [7], or Varnish [8], this paper focuses on improving this crucial component.

While many aspects of CDN optimization are not directly dependent on the nature of the content being served, as they are generally built in a content-agnostic manner, the application of "standard" CDNs to multimedia streaming delivery and, in particular, to Catch-up TV delivery is far from optimal, as this type of content exhibits a dynamic demand behavior that is not properly accommodated by traditional CDN caching algorithms [9], [10], [11].

Improving caching performance requires taking into consideration the underlying content demand patterns, and properly exploring them; therefore, this paper proposes a novel approach that takes advantage of content demand forecasts produced by a predictive machine learning model, built using Random Forests (RFs), to improve its caching decisions considering specific characteristics of the Catch-up TV content requested, i.e. in a content-aware manner. To ensure the soundness of the proposed approach and its results, Catch-up TV request logs are acquired from a popular Pay-TV service provider serving millions of users. The results show that content-aware approaches are suitable for significantly improving existing CDN caching nodes, and that their computational implementation cost is comparable to that of commonly used algorithms.

The remainder of this paper is organized as follows. Section 2 explores the current state-of-the-art in Catch-up TV caching algorithms, along with commonly employed caching strategies. Section 3 presents a detailed description of the proposed caching algorithm, whose performance evaluation is conducted in Section 4. Finally, Section 5 presents the concluding remarks.

2. Related Work

Catch-up TV is a key differentiating feature in modern Pay-TV services, whose popularity often surpasses other advanced time-shift features such as VoD and Digital Video Recorder (DVR) [1], [12]. As a consequence of its popularity, Catch-up TV imposes a severe strain on the delivery infrastructures, and has motivated researchers to tackle modeling and optimization challenges with the purpose of improving current delivery services and architectures. A common approach for delivery optimization is the usage of caching systems, which reduce the impact on backend servers and improve users' QoE.

The general issue of caching has been the subject of extensive research work, ranging from conceptually simple algorithms such as FIFO, LRU, and LFU [13], up to more advanced ones including LRU-K [14], LRU-HOT [15], and Low Inter-reference Regency Set (LIRS) [16]. LRU-K was developed to improve the caching performance of database buffers, while LRU-HOT's target is to keep "hot" items in cache, with the help of backend server-supported content flagging through HTTP MIME type headers, which prevent it from being easily deployed to practical solutions. As for LIRS, it improves LRU for content with weak locality; however, this assumption does not hold true for Catch-up TV. From these seminal works, Bélády's contribution [17] stands out by providing and demonstrating an optimal caching algorithm (MIN) still used today as a theoretical reference for the upper limit in achievable cache hit-ratios. In spite of this large research body, encompassing caching issues in many areas, there are a limited number of research studies that address Catch-up TV content caching.

The work of [18] stresses the big challenge of Catch-up TV caching, and investigates suitable strategies. A model is built that takes into account the evolution of content popularity, which is used by a caching algorithm that keeps track of the requests per item and dynamically builds the said model to estimate the relative importance of items and make caching decisions. The results show that this approach is able to outperform LRU and LFU for the 1.640 traces tested; however, the impact of the dynamic model building overhead is not considered in the performance simulations.

A complementary work is performed in [19], where Abrahamsson et.al provide an empirical IPTV work model based on a realistic scenario simulation which considers the large discrepancies in popularity, with the purpose of evaluating the performance of traditional caching algorithms, including LRU and LFU, and estimating the bandwidth requirements of time-shift services. The study's conclusions demonstrate that LFU is the most favorable caching approach; however, the study neglects the fact that Catch-up TV content has a life-time expectancy that must be taken into account, so that popular content that is no longer valid does not prevent new content from populating the caches. This research work is improved in [10], where additional effects are exploited, such as program popularity variability with time, and a characterization of its decay with time and genre. The results show that the content genre and the Catch-up TV availability window plays a very important role on the performance of caching algorithms, and therefore, on the streaming bandwidth required from the origin servers.

In summary, caching is a challenging proposition that has been widely researched and has a very significant impact on the overall performance of any data retrieval system. When applied to the context of Catch-up TV, a set of additional challenges arise; therefore, a new caching replacement approach is proposed and evaluated in the ensuing sections.

3. Most Popularly Used (MPU)

The proposal of new caching algorithms is usually motivated by 3 main goals: (i) improve the cache hit-ratio over competing solutions; (ii) reduce the algorithm's computational cost; (iii) lower the amount of data transferred from its original source to the cache. MPU focuses particularly on issues (i) and (iii), which are the most critical factors when delivering Catch-up TV content, provided that the associated computational costs remains within reasonable bounds.

MPU leverages content demand knowledge to make cache replacement decisions based on "priority maps". Priority maps are generated by online predictive machine learning algorithms, whose responsibility is to produce accurate content demand forecast for a given period. The predictive models are continuously improved by using past Catch-up TV consumption data. The generated priority maps contain enough information to unequivocally identify Catch-up TV items and their expected number of requests at each point in the future. The generation of priority maps should be performed per-PoP, as content demand may exhibit content locality that must be taken into consideration. An example of how priority maps may be built is provided on section 4.

MPU cache eviction policy favors items that have a greater expected priority, in detriment of others with lower expected priorities. Considering that MPU strongly depends on the priority maps, it is of utmost importance that the predictive machine learning algorithms are adequately tuned and able to perform accurate forecasts.

In order to properly depict the inner workings of MPU, we assume that a cache system containing a list $\mathscr{C}$ exists capable of holding S elements, and that the items to cache are represented by the set $\mathscr{I} = \{i_1, i_2, i_3, i_4, i_5 ... i_n\}$ and have an associated numeric priority from the set $\mathscr{P} = \{p_1, p_2, p_3, p_4, p_5 ... p_n\}$, so that item i_1 has p_1 priority, and so forth. $\mathscr{H}$ is a counter registering the total number of hits, while $\mathscr{M}$ counts the total number of misses.

These steps summarize how MPU works when an item is requested:

1) If the item already exists in cache, it is returned to the caller, and the total hit count is incremented;
2) If an item does not exist in cache, a miss is registered and the item is fetched from the origin server so that it may be returned to the caller;
3) If the cache is full or if a newly fetched item has a priority higher than the item with lowest priority in cache, MPU removes the item with the lowest priority and inserts the new one.

TABLE 1. MPU CACHE REPLACEMENT POLICY SAMPLE

Iteration	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Request	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
Result	miss	miss	miss	miss	**hit**	miss	**hit**	miss	**hit**	miss	**hit**	miss	**hit**	**hit**	**hit**	**hit**	**hit**	miss	**hit**	**hit**
Page 1	7	7	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Page 2		0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Page 3			1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

The pseudo code of MPU is presented in algorithm 1.

Algorithm 1: Most Popularly Used Algorithm

Input: $\mathscr{I}$, $\mathscr{P}$
Output: $\mathscr{H}$, $\mathscr{M}$
For every item $i \in \mathscr{I}$, perform the following operations.
Case 1: if $i \in \mathscr{C}$ then :
*Checks if item i exists in cache, if so, increment the total hits;
$\mathscr{H} \leftarrow \Delta 1$;
Case 2: otherwise, if $i \notin \mathscr{C}$ then :
*New miss is registered and the item is fetched from the origin server;
$\mathscr{M} \leftarrow \Delta 1$;
Case 3: if $|\mathscr{C}| \geq S$:
*Cache is full. Checks if new item i has higher priority than lowest
*priority item in cache;
if $p_i > \mathscr{C}_{min(p)}$:
*Delete the item with lowest priority in cache ;
*Insert new item i in the cache $\mathscr{C}$;

To exemplify how MPU behaves, we consider a common reference test sequence $\mathscr{S}$ [20]:

$$\mathscr{S} = (7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1) \quad (1)$$

where the max cache size S is 3 elements. The order is essential to the performance of cache algorithms, which must know which elements to keep in cache to maximize the possibility of cache hits.

Assume that each item in sequence S has a priority given by $\mathscr{P}_{i:p} = \{[0 : 6], [1 : 4], [2 : 4], [3 : 3], [4 : 1], [7 : 2]\}$, which maps each item to its respective priority. Higher priorities map to higher expectations that an item will be requested in the future. The step-by-step results of applying MPU to this sequence are presented in table 1. The three initial items each lead to cache misses and fully populate the cache for future use. Overall, MPU achieves 11 page hits and 9 page misses for the reference string $\mathscr{S}$ considered, which matches the 11 page hits achievable by Bélády's optimal algorithm. In spite of this purely theoretical exercise, the results show that MPU holds the promise of outstanding performance. The validation of MPU's performance under realistic conditions is the focus of the next section.

4. Results and Discussion

Having explored the design decisions behind MPU, this section presents a performance evaluation on the proposed content-aware caching algorithm.

First, an initial description of the dataset used to perform this evaluation under realistic conditions is conducted. Then, the procedure by which the demand forecasts are obtained is detailed, as they are essential to MPU. Finally, the performance tests are conducted with the purpose of comparing MPU to reference caching approaches.

4.1. Testing Methodology

The tests are implemented in *R* [21] using RStudio [22], and run on a Virtual Machine (VM) with 2 Intel E5-2640v3 CPUs, and 64GB of RAM. Even though the performance of the models considered in this analysis are dependent on their actual implementations, the tests are all performed in identical conditions and use common libraries.

When pertinent, the 95% Confidence Interval (CI) is shown on the average values' curve and data points.

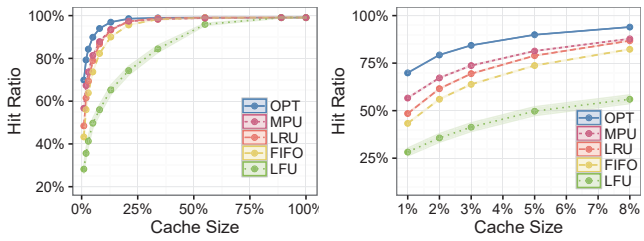
4.1.1. Dataset Description. A Catch-up TV consumption dataset is collected from a major Portuguese IPTV operator containing 30 days of program request logs, regarding the full month of April 2015. In total, the dataset contains over 22.505.901 unique requests, pertaining to 704.031 households and 866.720 different Set-Top-Boxes (STBs). Catch-up TV users had access to a total of 88.308 unique TV programs within the 30 days time period. Each log entry contains a request timestamp, user, and program metadata, which are used to build the forecasting models and traces for the algorithms performance validation.

4.1.2. Content Demand Forecasts and Testing Data. MPU relies on demand forecasts to make caching decisions; therefore, to conduct a performance evaluation it is necessary to build predictive models from the available dataset.

To create the demand forecast models, the request logs are split into two separate groups. The first 23 days are used as a training dataset for a Random Forest (RF) machine learning model which relies on regression techniques based on a forest of trees using random inputs [23] to predict the future demand of each available program. The *caret* package [24] is used to facilitate the model building procedure. As for the remaining 7 days worth of logs, from April 24 up to April 30, they represent the testing dataset and are used to create a sequential list of program requests which are the inputs of the caching algorithms.

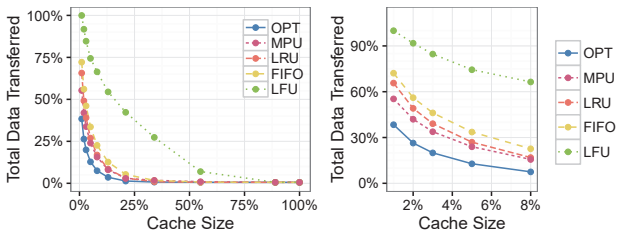
RFs are suitable predictive algorithms as they may be used in an "online" training mode, whereby an existing predictive model is improved using new data without requiring a full re-training process.

To facilitate the training of forecasting models and the generation of demand forecasts, the 7 days of test data are split into smaller time-slots, with a granularity of 1



(a) Full Results. (b) Zoom-In.

Figure 1. Hit Ratio vs. Cache Size.



(a) Full Results. (b) Zoom-In.

Figure 2. Backend Data Transfers vs. Cache Size.

hour. Therefore, for each program available in the testing period, $7 * 24 = 168$ demand forecasts are computed, which represent the expected number of requests, per program, in any given test time-slot.

4.1.3. Reference Cache Algorithms. In addition to testing MPU, reference caching algorithms, LFU, LRU, and FIFO, are implemented and serve as a comparison base for MPU's performance. Even though many other caching algorithms exist, most are either variations or combinations of the aforementioned algorithms. Furthermore, to understand the upper limit of achievable hit-ratio performance, Bélády's optimal page replacement algorithm (OPT) [17] is also implemented. The algorithms' core implementations are kept as similar as possible.

4.1.4. Key Performance Metrics. The performance tests focus on 3 key metrics:

- *Hit-ratio*: the ratio between the number of *hits* and the number of program requests. An indicator of how good the caching algorithm is on guessing programs that will be requested in the near-future;
- *Run time*: time required to run caching algorithms' code, the lower the better;
- *Backend data transfer*: estimated amount of data transferred from the *origin* server to the cache. As the purpose of caches is to reduce the impact on backend servers, a low metric is indicative of good caching performance.

The *run time* and *backend data transfer* results are presented in a normalized fashion, ranging from 0% to 100%, to facilitate the graphical analysis.

4.1.5. Cache Sizing. In order to explore the effect of different cache sizes in the performance of each algorithm, and the associated cost-benefit trade-offs, the parametric evaluations conducted in the *Performance Evaluation* section size the caches as fractions of the total number of unique available programs. Therefore, a cache size of 100% corresponds to a cache with the ability to hold the entire content catalog available on the 7 days testing window. To simplify the caches' implementation, each program is assumed to require 1 storage unit.

4.1.6. Bandwidth Calculations. Because the actual bandwidth requirements depend on factors such as video codec and resolution, to name a few, a dimensionless approach is taken that assumes that each program requires the transfer of 1 storage unit every time a cache miss occurs.

4.2. Performance Evaluation

Having established the test methodology, this subsection deals with conducting and discussing the performance results of the proposed caching algorithm. First a trade-off analysis is conducted that explores the impact of cache size on the performance metrics described in 4.1.4. Then, a time-varying perspective is provided which explores how the said key metrics evolve during the 7 days testing period described in 4.1. This time-varying analysis is essential as it allows an evaluation of the steady-state performance of each algorithm.

4.2.1. Hit Ratio vs. Cache Size. This analysis explores the impact of different cache sizes in the overall caches' hit-ratios. The results are presented in figure 1.

Starting with 1(a) it is possible to observe that the optimal (OPT) always provides the best performance, which is to be expected, while MPU performs much better than traditional caching algorithms. LRU performs worse than MPU but is much better that LFU and FIFO strategies. The algorithms' performance converges for cache sizes greater than 25%; however, we argue that this is not a common realistic scenario, which is mostly focused on cache sizes smaller than 10% of the overall corpus.

To better analyze this region, figure 1(b) presents a zoomed-in plot of the same results, where a clearer comparative study may be conducted. In this figure, it is possible to observe that, for cache sizes of 1%, MPU provides a hit-ratio 17% higher than LRU, the best performing traditional caching algorithm. The results show that MPU may be used to either lower the caches' sizes, for a given target hit-ratio, or to improve the cache hit-ratios for fixed storage sizes.

4.2.2. Backend Data Transfers vs. Cache Size. The volume of backend data transfers is a crucial metric that determines the scalability of CDNs by lowering the costly bandwidth requirements on the origin servers.

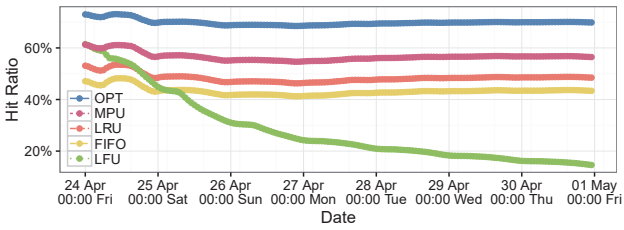


Figure 3. Hit Ratio vs. Time.

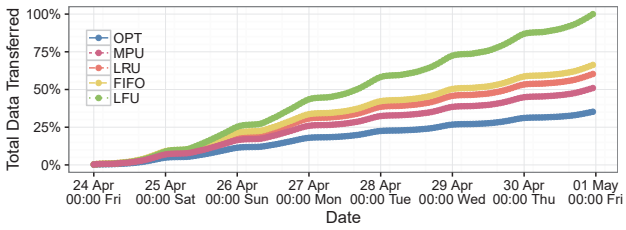


Figure 4. Backend Data Transfers vs. Time.

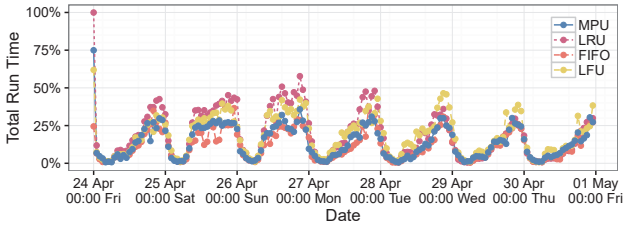


Figure 5. Cache Run Time vs. Time.

Figure 2 explores how this metric varies with the cache size. As in the previous analysis, the results are split into a global perspective, on figure 2(a), and on a zoomed-in plot presented in figure 2(b) which focuses on caches smaller than 10% of the total content corpus.

It is possible to observe that lower cache sizes correspond to higher data transfer requirements. These results illustrate the impact of cache misses, which require fetching the requested content from the origin server. Lower cache sizes translate into lower hit-ratios, as seen in figure 1, and on more cache *misses*. OPT provides the best performance, followed by MPU, LRU, FIFO, and LFU. For cache sizes of 1%, MPU requires 19% less backend data transfers than LRU, the next best performing cache algorithm. These results clearly demonstrate the potential of MPU in improving the scalability of CDNs solutions.

4.2.3. Hit Ratio vs. Time. Exploring how the caches' hit-ratios evolve with time is essential in Catch-up TV services where content popularity changes with time, and knowing the steady-state performance of caching nodes is a requirement. In order to perform this analysis the cache sizes are set at 1% of the total program corpus, which was previously shown to be a data point providing a cost-benefit trade-off where good caching performance is achievable with less than an order of magnitude of the total content.

Figure 3 presents the time-varying hit-ratio results for each caching algorithm where it is possible to observe that, as time progresses, some algorithms adapt better than others to content requests. Starting with the ideal algorithm, OPT, it provides the best overall caching performance, which is kept approximately constant with 70% hit-ratios. MPU provides the next-best result, with a significant performance advantage over LRU and FIFO. In spite of the different hit-ratios of MPU, LRU, and FIFO, their overall hit-ratios' curves behavior is similar and stabilize after the first day; thus, providing a consistent steady-state performance.

As for LFU, in spite of the excellent results for the early hours of day 24, its hit-ratios' performance progressively diminishes with time, which might be explained by the effect of "cache pollution", whereby items that were initially highly popular, but lose relevance, prevent other newer items from populating the caches; hence, leading to low hit-ratios.

The small increase in hit-ratios on all algorithms in day 24 is believed to be due to accentuated users' demand for popular content in some times of the day, i.e. a result of the *superstar* effect which also happens in the remaining days, albeit at a smaller scale.

4.2.4. Backend Data Transfers vs. Time. As a complement to the previous study, this analysis focuses on the evolution of backend data transfers with time, which are expected to evolve inversely proportionally to the hit-ratios of each solution. As in the previous analysis, the cache sizes are set at 1% of the overall content corpus.

Figure 4 shows that while every caching algorithm starts with approximately the same amount of data transferred, as time progresses they quickly diverge, especially LFU, which provides the worst overall results at the end of day 30. MPU performs significantly better than LRU and FIFO, by transferring 18% less data than LRU.

As in the previous results, the daily demand changes are perceptible with slight variations in the transferred data curve for each caching algorithm.

4.2.5. Cache Run Time vs. Time. The final evaluation is centered around the evolution of cache run time with time. The cache sizes are set at 1%. This is an important metric, as the high-performance requirements of CDNs constrains the selection of caching algorithms to those that are computationally efficient and scalable. Figure 5 presents a graphical analysis on how the computational requirements of each caching algorithm varies with time. OPT is excluded from the results as it is not implementable in practice.

It is possible to observe that all caching algorithms exhibit a similar behavior with respect to their computational requirements, even though some algorithms do require more processing time than others. MPU and FIFO are the least computationally demanding caching algorithms, while LFU and LRU require more time to perform their tasks. The initial observed run time peak for every caching strategy is due to the caches' warm-up process.

5. Conclusion

Multimedia delivery in OTT environments is particularly challenging, particularly in the case of Catch-up TV content with its dynamic demand patterns that make it hard for traditional caching algorithms to exhibit, and sustain, high performance levels.

To address the issues with Catch-up TV caching in OTT environments, a novel algorithm, MPU, is proposed that is able to leverage content demand forecasts to provide significantly better cache performance metrics. The results show that the use of MPU enables either significant cache costs savings, for a fixed target hit-ratio, or much better caching performance when run using identical storage resources.

Future improvements will focus on adding scan-resistance and on exploring the algorithm's robustness when faced with unknown content.

Acknowledgments

The authors would like to thank Fausto Carvalho (Altice Labs, SA) and João Ferreira (MEO - Serviços de Comunicações e Multimédia, SA) for the key discussions and for providing the raw Catch-up TV dataset.

This research work was funded by the UltraTV project (Portugal 2020 POCI-01-0247-FEDER-017738), by FCT/MEC through national funds, and, when applicable, co-funded by FEDER PT2020 partnership agreement under the project UID/EEA/50008/2013 OT2Delivery (Over-the-top Multimedia Content Delivery for Next Generation Mobile Networks).

References

[1] Nielsen, "The Digital Consumer," pp. 1–28, 2014, Accessed: 09-2015. [Online]. Available: http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2014Reports/the-digital-consumer-report-feb-2014.pdf

[2] CNC, "L économie de la télévision de rattrapage en 2014," *Centre national du cinéma et de limage animée*, pp. 1–33, 2015. [Online]. Available: http://www.cnc.fr/web/fr/etudes/-/ressources/6592632

[3] J. Abreu, V. Becker, J. Nogueira, and B. Cardoso, "Time-shift services : a taxonomy and techno-business impacts of Catch-up TV," in *CENTERIS 2015 - Conference on ENTERprise Information Systems / PRojMAN 2015 - International Conference on Project MANagement / HCIST 2015 - International Conference on Health and Social Care Information Systems and Technologies*, 2015, p. 6.

[4] J. Famaey, F. Iterbeke, T. Wauters, and F. De Turck, "Towards a predictive cache replacement strategy for multimedia content," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 219–227, 2013.

[5] T. S. Bjøndal and M. Gedde, "Ubiquitous TV: A Business Model Perspective on the Norwegian Television Industry," Master, Norwegian University of Science and Technology, 2011. [Online]. Available: http://brage.bibsys.no/xmlui/handle/11250/266027

[6] The Apache Software Foundation, "Apache Traffic Server," 2015, Accessed: 12-2015. [Online]. Available: http://trafficserver.apache.org

[7] NGINX Inc., "NGINX High Performance Web Server," 2015, Accessed: 12-2015. [Online]. Available: http://nginx.com

[8] Varnish Software, "Varnish Cache," 2015, Accessed: 12-2015. [Online]. Available: https://www.varnish-software.com/

[9] J. Nogueira, L. Guardalben, B. Cardoso, and S. Sargento, "Catch-up TV Analytics: Statistical Characterization and Consumption Patterns Identification on a Production Service." *Multimedia Systems*, 2016. [Online]. Available: http://dx.doi.org/10.1007/s00530-016-0516-7

[10] H. Abrahamsson and M. Bjorkman, "Caching for IPTV distribution with time-shift," in *2013 International Conference on Computing, Networking and Communications (ICNC)*. San Diego, CA: IEEE, Jan. 2013, pp. 916–921. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6504212

[11] G. Nencioni, N. Sastry, J. Chandaria, J. Crowcroft, S. Nishanth, J. Chandaria, and J. Crowcroft, "Understanding and Decreasing the Network Footprint of Catch-up TV," in *Proceedings of the 22Nd International Conference on World Wide Web*. Rio de Janeiro, Brazil: International World Wide Web Conferences Steering Committee, 2013, p. 12. [Online]. Available: http://dl.acm.org/citation.cfm?id=2488388.2488472

[12] ANACOM, "Subscription Television Service Statistical Information 2nd Quarter 2015," ANACOM, Tech. Rep., 2015, Accessed: 12-2015. [Online]. Available: http://www.anacom.pt/streaming/STVS2quarter2015.pdf?contentId=1366508&field=ATTACHED_FILE

[13] A. Balamash and M. Krunz, "An overview of web caching replacement algorithms," *IEEE Communications Surveys & Tutorials*, vol. 6, no. 2, pp. 44–56, 2004. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5342239

[14] E. J. O'Neil, P. E. O'Neil, and G. Weikum, "The LRU-K page replacement algorithm for database disk buffering," *ACM SIGMOD Record*, vol. 22, no. 2, pp. 297–306, Jun. 1993. [Online]. Available: http://portal.acm.org/citation.cfm?doid=170036.170081

[15] J.-M. Menaud, V. Issarny, and M. Banâtre, "Improving the Effectiveness of Web Caching," in *Advances in Distributed Systems: Advanced Distributed Computing: From Algorithms to Systems*, S. Krakowiak and S. Shrivastava, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 375–401. [Online]. Available: http://dx.doi.org/10.1007/3-540-46475-1_16

[16] S. Jiang and X. Zhang, "LIRS," *ACM SIGMETRICS Performance Evaluation Review*, vol. 30, no. 1, p. 31, jun 2002. [Online]. Available: http://portal.acm.org/citation.cfm?doid=511399.511340

[17] L. a. Belady, "A study of replacement algorithms for a virtual-storage computer," *IBM Systems Journal*, vol. 5, no. 2, pp. 78–101, 1966. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5388441

[18] Z. Avramova, D. De , S. Wittevrongel, and H. Bruneel, "Performance analysis of a caching algorithm for a catch-up television service," *Multimedia Systems*, vol. 17, no. 1, pp. 5–18, Aug. 2011. [Online]. Available: http://link.springer.com/10.1007/s00530-010-0201-1

[19] H. Abrahamsson and M. Bjorkman, "Simulation of IPTV caching strategies," in *Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2010 International Symposium on*. Ottawa, ON: IEEE, Jul. 2010, pp. 187–193. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5588896

[20] A. Silberschatz, P. B. Galvin, and G. Gagne, *Applied Operating Systems Concepts*. Wiley, 1999. [Online]. Available: https://books.google.pt/books?id=rFM0cSsElYgC

[21] R Foundation for Statistical Computing, "The R Project for Statistical Computing," 2016, Accessed: 01-2016. [Online]. Available: https://www.r-project.org/

[22] RStudio Inc., "RStudio," 2016, Accessed: 01-2016. [Online]. Available: https://www.rstudio.com/

[23] A. Liaw, "randomForest," 2015, Accessed: 09-2015. [Online]. Available: https://cran.r-project.org/web/packages/randomForest/randomForest.pdf

[24] M. Kuhn, "caret," 2015, Accessed: 09-2015. [Online]. Available: https://cran.r-project.org/web/packages/caret/caret.pdf

Appendix G

Content-Aware Over-The-Top Delivery of Catch-up TV Services.

To be submitted to the *IEEE's Transactions on Multimedia.*

Content-Aware Over-The-Top Delivery of Catch-up TV Services

João Nogueira, André Dias, Lucas Guardalben, Bernardo Cardoso, and Susana Sargento

***Abstract*—The migration of resource-intensive Catch-up TV services from managed IP Television (IPTV) infrastructures to Over-The-Top (OTT) delivery is essential for achieving a convergent and cost-effective anytime-anywhere solution.**

Content Delivery Networks (CDNs) are widely used as the backbone for scalable OTT delivery solutions; however, they are often tailored for generic web content and do not consider the particularities of Catch-up TV content, namely its dynamic popularity behavior, lifetime expectancy, and superstar effects, to name a few, as shown in the existing scientific literature.

To address these shortcomings and enable an efficient delivery of Catch-up TV services, this research work proposes, discusses, and provides an experimental evaluation of a content-aware delivery approach capable of leveraging online machine-learning techniques to predict the users' requirements and simultaneously optimize the performance of delivery systems taking into consideration the content's characteristics and demand patterns.

Experimental tests show that existing commercial solutions extended with the proposed approach exhibit significant performance gains in terms of average request latency, cache hit-ratios, backend bandwidth demand and users' QoE.

***Index Terms*—Catch-up TV, IPTV, OTT, CDN, Content-aware**

I. Introduction

Large scale delivery of Catch-up TV content is a challenge faced by IPTV operators, struggling to cope with a service demand that keeps growing and is already much larger than that of traditional Video-on-Demand (VoD) [1].

This struggle is particularly relevant for operators transitioning to Over-The-Top (OTT) delivery with the goal of achieving the *anytime-anywhere* promise of convergent solutions while lowering the overall Capital Expenditures (CAPEX) and Operational Expenditures (OPEX) requirements of this resource-intensive multimedia service [2].

Given the popularity of OTT multimedia traffic on the Internet, which is expected to reach over 80% of its overall traffic by 2019 [3], an exploration of optimization opportunities is required. To address scalability challenges in OTT scenarios, a common approach is to employ CDNs to get content close to users, improve their Quality-of-Experience (QoE), and minimize network traffic costs; however, the use of CDNs is rife with difficulties. It has been shown that, when used to deliver Pay-TV services, CDNs may suffer from low caching performance or introduce excessive delays if not carefully tuned [4]. The application of "standard" CDNs to multimedia streaming delivery and, in particular, to Catch-up TV delivery is far from optimal, as this type of content exhibits a dynamic demand behavior that is not properly accommodated by traditional CDN replica servers [5]. A CDN should not be agnostic to its content so that better performance levels are achieved, hence the need for *content-aware* CDNs. *Content-awareness* refers to the adaptation of data storage, processing or transmission methods according to characteristics of the content being delivered, and is highly dependent on the systems' ability to extract meaningful information from it.

Considering these issues, and the fact that the overall performance of CDNs is highly dependent on the efficient usage of the available servers, measured in computational, memory, and network requirements, this research work proposes, details, and evaluates a content-aware caching architecture capable of leveraging demand forecasts produced by predictive machine learning models to provide dynamic resource allocation capabilities, while simultaneously improving caching decisions considering specific characteristics of the requested content.

To validate and ensure the soundness of the proposed approach, Catch-up TV request logs are acquired from a popular Pay-TV service provider serving millions of users. The results show that the presented content-aware approach is suitable for significantly improving existing CDNs.

In summary, this work provides the following contributions:

- Proposal of a novel content-aware OTT delivery architecture with a detailed discussion and modeling of its building blocks', features and responsibilities;
- Proposal of a prediction algorithm based on machine learning to forecast Catch-up TV programs requests;
- Proposal of an advisor algorithm that decides on the distributed caching configuration to optimize CDN performance with cache size minimization;
- Experimental implementation of the proposed architecture targeting a Catch-up TV delivery use-case;
- Performance validation of the content-aware delivery architecture using requests logs from a production Catch-up TV service, considering key Quality-of-Service (QoS) metrics and QoE estimations.

The remainder of this paper is organized as follows. Section II provides a literature review on the relevance of content-awareness and its applicability to CDNs, in addition to reviewing pertinent research work on caching algorithms, dynamic and autonomic cloud resource management, and Catch-up TV services' characterization. The proposed content-aware OTT delivery architecture is presented in detail on Section III, while Section IV describes how the experimental validation is conducted. The results are presented on Section V, followed by the concluding remarks on Section VI.

J. Nogueira is with Altice Labs SA, 3810-106 Aveiro, Portugal; University of Aveiro, 3810-193 Aveiro, Portugal; and Instituto de Telecomunicações, 3810-193 Aveiro, Portugal (e-mail: joaonogueira@ua.pt).

A. Dias, L. Guardalben and S. Sargento are with University of Aveiro, 3810-193 Aveiro, Portugal; and Instituto de Telecomunicações, 3810-193 Aveiro, Portugal (e-mail: andre.dias@ua.pt, guardalben@ua.pt, susana@ua.pt)

B. Cardoso is with Altice Labs SA, 3810-106 Aveiro, Portugal (e-mail: bernardo@alticelabs.com)

This research was supported by grants to P2020 UltraTV (POCI-01-0247-FEDER-017738) and OT2Delivery (UID/EEA/50008/2013) projects.

II. Related Work

The importance of leveraging content-specific information has been identified by the scientific and industrial community as an effective way of improving new and existing systems. These concepts are applicable to a wide range of research areas; however, one in particular has received the most attention: Future Internet Applications and their Information-Centric Networking (ICN) components [6], [7].

In [8], an all-encompassing approach is taken to simultaneously solve the problems of request routing, node placement, and content eviction. The authors abstract the CDN as a switch-scheduling problem and propose 3 different algorithms inspired on the Max-Weight scheduling algorithm, whereby content popularity is inferred by analyzing the request queues. In this case, content-awareness refers to the fact that each source is aware of every item held by the caches, which poses a distributed knowledge synchronization problem that is not easily solvable for large, heterogeneous, CDNs.

The work in [9] proposes a multi-criteria optimization algorithm in scenarios where information arrives from multiple sources, with the purpose of jointly optimizing the selection of the best delivery server and path. The issue is presented along with the definition of what is an efficient solution. Substantial gains are shown by applying the proposed criteria; however, the baseline comparison relies on random server selection, which is not representative of commercial delivery solutions.

Mangili *et al.* [10] focus on the issue of network planning, with the purpose of modeling and studying the migration to future ICNs. Using a Mixed Integer Linear Programming (MILP) formulation, their findings suggest that the migration of a small set of agnostic nodes to content-aware ones is enough to provide substantial traffic reduction benefits.

The authors of [11] propose a content-aware dynamic load-balancing algorithm capable of taking into account not only the servers' load, capabilities, queue lengths, and historical performance, but also content characteristics, regarding their computational and bandwidth impacts. The approach is shown to significantly outperform static load-balancing algorithms (weighted Round-Robin) in terms of content response delay.

Considering these research works, it is clear that the performance and implementation of content-aware systems are highly dependent on the available information's quality.

As this study's focus is on Catch-up TV, it is important to review existing models that accurately describe Catch-up TV content and enable a thorough understanding of how, when, and what Catch-up TV programs are demanded by users.

In [12], the authors characterize Catch-up TV users' behaviors, such as the duration of viewing sessions, genre preferences, and program popularity analysis, to name a few. The presented statistical analyses support the existence of dynamic popularity and consumption patterns, depending on the time at which the service is used, and also on content characteristics such as its original airing time, date and genre.

The study conducted by Beauvisage *et al.* [13] points to a contradiction of the *long-tail* hypothesis, in favor of the *superstar* effect whereby a small fraction of the available programs receive the vast majority of user requests, while also showing that users favor recently aired programs in detriment of older ones. Similar results were attained by [5], which adds that users overwhelmingly prefer serialized content.

Another service providing Catch-up TV content is thoroughly studied in [14]. Its conclusions point to the occurrence of the Pareto-principle, whereby the 20% most popular assets are responsible for 80% of the total content requests.

These characterization and modeling studies hint at significant optimization opportunities that can be achieved by content-aware Catch-up TV CDNs.

Regarding the *cacheability* of Catch-up TV content, new caching algorithms have been proposed [15], along with derivations of Least Recently Used (LRU) and Least Frequently Used (LFU) [16], but none has been able to reach performance levels comparable to that of Bélády's optimal caching algorithm [17], still used today as a theoretical reference for the upper limit in achievable cache hit-ratios.

In addition to having the potential to significantly improve the performance of caching algorithms, content-aware delivery technologies enable the development of other smart-CDN components, such as smart maintenance scheduling and dynamic resource provisioning systems, that are capable of adjusting allocated resources to the expected user demand based on forecasts derived from content-specific knowledge.

The problem of dynamic and autonomic cloud resource management has been explored by several authors. The work in [18] provides an overview of open issues on cloud resource orchestration, while stressing the difficulties associated with dealing with pervasive, highly dynamic and heterogeneous cloud computing resources requiring expert knowledge for deployment, maintenance, monitoring, and control tasks.

The work in [19] identifies the need for dynamic network resource provisioning as essential to maintaining a high-QoE in entertainment systems. The authors propose the inclusion of a management and control plane responsible for a prediction engine, combining long and short-term forecasts for resource utilization which are used to decide the optimal delivery approach, such as using CDNs, or engaging in Peer-to-Peer (P2P) distribution. In [20], a survey is conducted on forecasting and profiling models, which frames the relevance of the problem at hand and systematizes the key motivations behind these techniques, namely application, resource, and cost management. Autonomic resource management is well represented by the MAPE-K (Monitor, Analyze, Plan, Execute, Knowledge) autonomic loop [21], and its related *self-** challenges.

The following work is focused on content-aware delivery optimization of Catch-up TV services; however, the proposed concepts and methods are not limited to this use-case and may be generally applicable to other content-delivery scenarios.

III. Proposed Content-Aware Over-The-Top Delivery Architecture

Having considered the potential benefits of content-aware approaches to improve delivery systems, along with key Catch-up TV characteristics that are essential to the design of an optimized OTT delivery solution, this section proposes a new architecture that maximizes the performance of Catch-up TV OTT CDNs through content-aware mechanisms.

Figure 1 exhibits the envisioned global architecture along with its main components. A macro overview of the proposed architecture presents 6 different functional blocks. The *Catch-up TV Content Origin* is responsible for holding the complete set of Catch-up TV content, the associated metadata, and

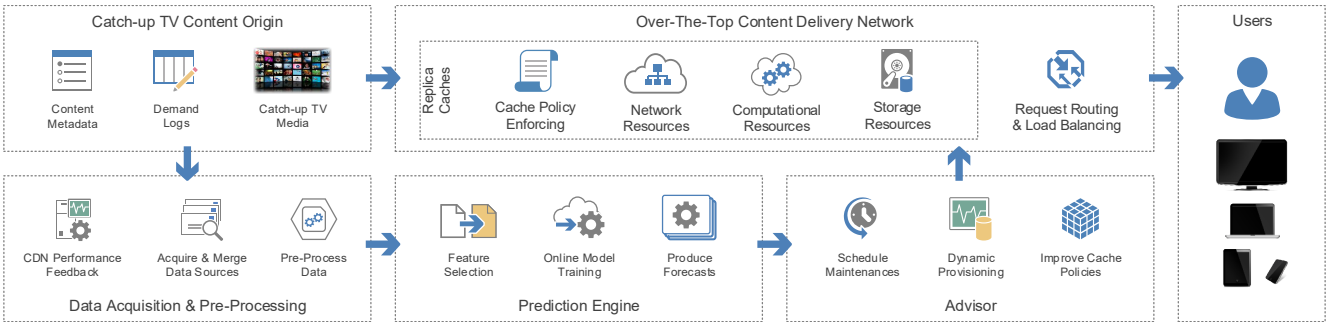


Fig. 1. Proposed Content-Aware Over-The-Top Catch-up TV Delivery Architecture.

user request logs. Next, the *Over-The-Top Content Delivery Network* block represents the actual system responsible for the efficient and high-QoE delivery of Catch-up TV content, through the use of replica caches to the *Users*, which are the final consumers of the Catch-up TV service.

The remaining 3 functional blocks are responsible for ensuring an optimal operation of the *Over-The-Top Content Delivery Network*. The purpose of the *Data Acquisition & Pre-Processing*, *Prediction Engine*, and *Advisor* blocks is to recommend maintenance schedules, create and distribute dynamic provisioning and caching policies to be used by the *Over-The-Top Content Delivery Network*. By working as a complement to the main content delivery flow, this architecture enables non-disruptive improvements to current CDN solutions. The detailed responsibilities of each individual element and sub-elements is provided in the ensuing subsections.

A. Catch-up TV Content Origin

This component, commonly known as *Origin*, aggregates three main responsibilities:

- *Content Metadata* contains information that is associated with each media element. In Catch-up TV services, content metadata includes Electronic Programming Guide (EPG) information, such as the original program airing date, broadcast station, program title, episode number, series identifier, and duration, to name a few;
- *Demand Logs* provide traceability by recording *who* requested *what* and *when*, therefore providing timestamped records that map user requests to Catch-up TV programs;
- *Catch-up TV Media* is the actual media vault responsible for holding the encoded media elements, usually also encrypted, ready to be delivered to end-users.

In practice, the *Origin* may also have to interface with external Business Support Systems (BSSs) and Operations Support Systems (OSSs), but these are its main responsibilities.

B. Over-The-Top Content Delivery Network

This block is the optimization target of the overall content-aware architecture. It is often composed of multiple servers, called the replica, surrogate, or cache servers, and is responsible for delivering Catch-up TV content from the *Origin* to the end-users. The replica servers are interconnected and store content copies to reduce the load on *Origin* servers and network interconnect, while also increasing the services' QoE.

They may be characterized by their *Computational*, *Storage*, and *Network* resources, which should be adequately dimensioned taking into the consideration the services' QoE vs. OPEX/CAPEX trade-off. Given that it is often not economically viable to fully replicate the *Origin's* content, replica servers must employ caching strategies to carefully select what to keep in storage and what to discard, thus *Cache Policy Enforcing* is a key function of replica servers.

In addition to the replica servers, *Request Routing & Load Balancing* systems are also required to properly direct users and traffic to the most suitable replica servers.

C. Users

The *Users* element represents the services' consumers. They may be geographically dispersed and use any Internet-connected device to access Catch-up TV content on-demand. It is important to properly model the users' demand profiles to adequately tune and dimension the CDNs' resources, i.e. network, storage, and computing.

D. Data Acquisition & Pre-Processing

The *Acquire & Merge Data Sources* element interfaces with data-sources that contain relevant information regarding the content being cached and merges it into meaningful representations. For Catch-up TV, suitable data-sources include the EPG, *Content Metadata*, analytics events providing information regarding users' requests and preferences, i.e. *Demand Logs*, as well as *CDN Performance Feedback* metrics.

A meaningful data representation maps a set of user requests to a specific TV program, accompanied by its metadata — such as its original airing date, TV station, etc. — along with prior CDN performance metrics for that particular content.

Past performance metrics create a feedback loop that aids the accuracy of future predictions by providing information regarding past prediction errors.

After the initial data acquisition and merging process, *Pre-Processing* is applied in order to compensate for discrepancies caused by the predictors' different scales, standard deviations, and average values. These discrepancies in scale and statistical properties often impair the numerical stability and bias of learning algorithms, potentially favoring some predictors over others, not because of their real importance but because of their different scales and distributions; therefore, it is important to scale, center, and correct the skewness – e.g. using a Yeo and Johnson transformation [22] – of each predictor before making the data available to the *Prediction Engine*. These transformations are easily performed using free and open-source software, such as R's *caret* package [23].

The complexity of the data gathering procedures varies with the actual production environments, as sophisticated data-gathering systems may have to be employed to gather the relevant data from multiple sources.

E. Prediction Engine

The prediction engine is key in this content-aware approach, and it is where the learning and forecasting cores of the content-aware caching solution are implemented. Its responsibility is to gather inputs from the *Data Acquisition & Pre-Processing* component and to generate accurate predictions regarding future Catch-up TV programs' requests that influence the CDN's configuration and overall performance. Depending on the available data and topology, the module may be required to predict consumer demand per Point-of-Presence (PoP).

A mathematical description is hereby presented to clarify the operations performed by *Prediction Engine*'s components.

P represents the set of p unique Catch-up programs, S comprises the set of s available predictors that describe each log entry, – containing program, user, and CDN performance data, as described in the previous sections –, and L is matrix of log entries, with m rows, and $|S|$ columns.

We define t as a timestamp variable, measured since the epoch (1970-01-01 00:00:00 UTC), in hours – Equation 1. Empirical findings and prior data analysis [12], [15] indicate that 60 minutes time slots represent an adequate compromise between time precision and computing requirements, even though specific scenarios may require a better time resolution.

$$t = \left\lfloor \frac{\Delta t_{epoch}}{3600} \right\rfloor \tag{1}$$

1) Feature Selection: To generate demand forecasts per program, this block ingests data from the *Data Acquisition & Pre-Processing* component, leveraging supervised and unsupervised techniques to perform an initial selection of predictors. Supervised methods rely on previous data and known outcomes, while unsupervised approached do not.

In this work, a supervised *filter* method is employed based on *ensemble* selection, implemented in R's *fscaret* package [24]. Unsupervised selection is performed through Near-Zero Variance (NZV) and cross-correlation analyses [23]. The feature selection process takes into account the fact that a Catch-up TV program must be unequivocally identifiable using a minimum set of predictors. Equation 2 illustrates the log data matrix L, with m log entries and $|S|$ predictors.

$$L = \begin{pmatrix} l_{11} & \dots & l_{1n} \\ \vdots & \ddots & \vdots \\ l_{m1} & \dots & l_{mn} \end{pmatrix} = (l_{in}) \in \mathbb{R}^{m \times n} : n = |S| \tag{2}$$

The filtering process selects a subset $S' \subset S$ of predictors as a result of the individual techniques. S' is presented on Equation 3, where S_S represents the set of filtered predictors using supervised methods, and S_U using unsupervised methods.

$$S' = S_S \cup S_U : S' \subset S \tag{3}$$

As a result of the filtering process, the final set of log data to be used in the subsequent steps relies only on the filtered S' predictors, so that matrix L' contains the same number of log entries as L, but with $|S'|$ predictors only – Equation 4.

$$L' = (l_{mn}) \in \mathbb{R}^{m \times n} : n = |S'| \tag{4}$$

2) Online Model Training: After defining the forecasting constraints and time-granularity decisions, the *Online Training Model* block leverages the selected predictors S' to retrain a Random Forest (RF) machine learning algorithm [25], using the filtered matrix L'. The retraining function is illustrated in Equation 5, denoted by $T()$, whose parameters are M_t – the latest forecasting model at time t – and L'_{t+1} – the newly filtered data matrix. As a result, the training function generates a new forecasting model M_{t+1}, which will be used on the forecasting step. Even though other regressive algorithms might be employed, as long as online model training is supported, our previous findings suggest that RFs have good predictive capabilities for this particular use-case.

$$T(M_t, L'_{t+1}) = M_{t+1} \tag{5}$$

On the following step, the updated forecasting model M_{t+1} will be used to *Produce Forecasts* for each program expected to air in the period under analysis.

3) Forecasting: The final process is to *Produce Forecasts* for each program p expected to air in the period under analysis, e.g. $t+1$, with detailed demand predictions for each program. Equation 6 presents a forecasting function $F()$ taking as input the latest forecasting model M_t, the target forecasting time slot $t+1$ and program p, and outputting a program demand estimate for the desired period.

$$F(M_t, t+1, p) = D_{p(t+1)} \quad \forall p \in P \tag{6}$$

As a result of the forecasting function, a full estimate on the upcoming program demand is achieved, and the generated forecasts are pushed to the *Advisor*, whose purpose is to manage and distribute configurations to replica CDN nodes.

F. Advisor

Demand forecasts produced by the *Prediction Engine*, D_{pt}, are leveraged by the *Advisor* in three different manners.

First, from an operational perspective, knowing when users' demand is the lowest helps to optimally *Schedule Maintenance*, such as running software updates, file-system checks, or other operations that would be undesirable when the systems are heavily loaded, in order to prevent a negative QoE impact.

A scheduling algorithm example is described by Equation 7, which picks the best maintenance time slot according to the expected total demand, within a given time-window W defined for a set of possible time slots $\{t, t+1, \dots\} \in W$.

By providing additional constraints, such as PoP location, this formulation may be trivially expanded to produce maintenance schedules specific to individual PoPs.

$$\text{minimize} \quad \sum_{p \in P} D_{pt} \qquad \text{subject to} \quad t \in W \tag{7}$$

Second, from a cost optimization perspective, accurate forecasts enable aggressive dynamic resource provisioning policies, where significant power, computational, bandwidth, and storage savings are possible without compromising the services' performance and users' QoE. From the D_{pt} forecasts it is possible to predict the amount of required storage and bandwidth, given that they depend directly on the characteristics of program p and its demand at time t.

Finally, due to the detailed knowledge on future demand, the *Advisor* is also responsible for acting as a coordination agent for distributed caching configurations with the purpose of optimizing replicas' caches. We proposed Most Popularly Used (MPU) [15] to explore the program demand forecasts D_{pt} to intelligently select the subset $P' \subset P$ specifying the p programs to hold in cache at each time slot t.

MPU cache eviction policy favors items that have a larger expected priority, in detriment of others with lower expected priorities. Considering that MPU strongly depends on the priority of each content, it is of utmost importance that the predictive machine learning algorithms are adequately tuned and able to perform accurate forecasts.

In MPU – Algorithm 1 –, we assume that a cache system containing a list $\mathscr{C}$ exists capable of holding N elements, that the items to cache are represented by the set $\mathscr{I} = \{i_1, i_2, i_3, i_4, i_5 ... i_n\}$ and have an associated numeric priority from the set $\mathscr{P} = \{p_1, p_2, p_3, p_4, p_5 ... p_n\}$, so that item i_1 has p_1 priority, and so forth. $\mathscr{H}$ is a counter registering the total number of hits, while $\mathscr{M}$ counts the total number of misses.

When an item is requested, MPU:

1) Checks if item already exists in cache. If so, the item is returned and the total hit count is incremented;
2) If an item does not exist in cache, a miss is registered and the item is fetched from the origin server so that it may be returned to the caller;
3) If the cache is full or if a newly fetched item has a priority higher than the item with lowest priority in cache, MPU removes the item with the lowest priority and inserts the new one.

Algorithm 1: Most Popularly Used Algorithm

Input: $\mathscr{I}$, $\mathscr{P}$
Output: $\mathscr{H}$, $\mathscr{M}$
For every item $i \in \mathscr{I}$, perform the following operations.
Case 1: if $i \in \mathscr{C}$ then :
    *Checks if item i exists in cache, if so, increment the total hits;
    $\mathscr{H} \leftarrow \Delta 1$;
Case 2: otherwise, if $i \notin \mathscr{C}$ then :
    *New miss is registered and the item is fetched from the origin server;
    $\mathscr{M} \leftarrow \Delta 1$;
Case 3: if $|\mathscr{C}| \geq N$:
    *Cache is full. Checks if new item i has higher priority than lowest
    *priority item in cache;
    if $p_i > \mathscr{C}_{min(p)}$:
        *Delete the item with lowest priority in cache ;
    *Insert new item i in the cache $\mathscr{C}$;

In summary, the *Advisor* fine-tunes the configurations and available replica servers' resources with the purpose of simultaneously improving the users' QoE and reducing costs due to power, bandwidth, storage and computational savings.

IV. Experimental Validation

This section describes the implementation and testing procedures used to validate the proposed architecture relying on readily available solutions for OTT CDNs.

A. Dataset Description

The dataset's quality is critical for the performance of any forecasting algorithm. In this work, a Catch-up TV consumption dataset is collected from a major IPTV operator containing 30 days of program request logs, regarding the full month of April 2015 [12]. This nonlinear service provides free access to the previous 7 days of program airings, depending on users' subscriptions. Personal user details are anonymized. The key dataset information is summarized as follows: 22.505.901 requests with device, location, and EPG data; 704.031 households; 866.720 Set-Top-Boxes; 80 TV channels; 88.308 TV programs; full month of April 2015.

B. Training and Testing Data

The request logs are split into 2 different groups, according to their purpose. The first group, reserved for *training*, is comprised by the initial 23 days of logs, while the remaining 7 days are held up and used for performance assessment purposes. Considering the previously established time slot granularity of 1 hour and the testing period, a total of $7 * 24 = 168$ demand forecasts are computed.

C. Catch-up TV Content Origin

1) Demand Logs & Content Metadata: The training data set contains both the demand logs and the associated content metadata; therefore, these components get their information from the same data source.

2) Catch-up TV Media: For the media vault, a Microsoft Internet Information Services (IIS) server is set up with Smooth Streaming [26] content to mimic the expected OTT scenarios leveraging adaptive bitrate encoded content. For practical reasons, regardless of the Catch-up TV program requested, the same video content is always provided. When crafting the content request URL, the query strings are modified to ensure that the CDN treats each program independently.

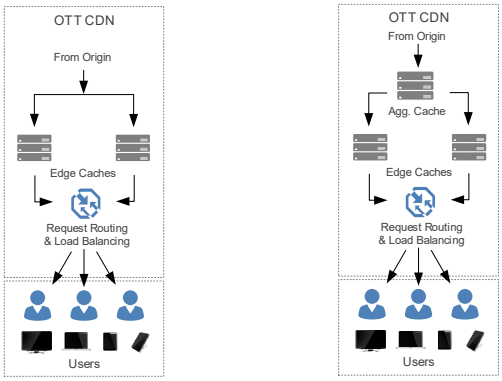
D. Over-The-Top Content Delivery Network

CDN architectures are suitable to *proxy-cache* deployments; therefore, the experimental validation focuses on replica cache solutions with 1 and 2-tier caching layers, i.e. with and without *Aggregation Caches*, as depicted in figures 2(a) and 2(b).

Adding aggregation caches to OTT CDNs is useful. In mobility scenarios where base station changes occur, content previously cached on an edge cache will likely be present on the local aggregation cache, avoiding costly requests to origin servers when the user changes from one edge cache to another neighbor one. Moreover, in the event of an edge cache failure, the aggregation cache helps with edge cache rebuilds that are faster and impose a lower strain on the origin server. It is mostly useful in scenarios with high geographical diversity and/or large delays to the origin.

Even though most common proxy-cache solutions, such as Nginx [27], and Squid [28] are open-source and modifiable, a choice was made to use Apache Traffic Server (ATS) [29] as the underlying framework for implementing the custom *Edge Caches* and *Aggregation Caches*. The reason for this choice was of practical nature, as this project's code is well documented and easy to extend.

The *Request Routing & Load Balancing* tasks are handled by HAProxy [30], which uses a Round-Robin strategy to randomly distribute requests within the Edge Caches. For co-located Edge Caches, a more suitable approach would be to use consistent URL hashing to ensure that identical requests are always processed by the same server and to avoid a full remapping in the event of a server failure. However, given that the experimental evaluation purpose is to simulate Edge Caches that may not be co-located it makes more sense to randomly distribute the clients' requests.



(a) 1-Tier Caching. (b) 2-Tier Caching.

Fig. 2. Experimental Replica Cache Architectures.

E. Users

Catch-up TV users are simulated through Python scripts performing HTTP requests to the load balancer. In order to ensure an accurate reproduction of real scenarios, the requests are performed sequentially, according to their original order in the previously described 7 days training request logs.

The ordered and predictable nature of users' requests ensures that different test runs produce comparable results.

F. Testbed Description

Tests are run in a virtualized environment (VMware ESXi 5.5.0), using an HP ProLiant DL160 Gen9 server with 2 x Intel E5-2640v3 CPUs (32 cores) and 32GB of RAM. The detailed resource reservations per component are presented on Table I. An additional identical server is connected to the management network and is used for the *Data Aquisition and Pre-Processing*, *Prediction Engine*, and *Advisor* tasks.

	Load Balancer	Edge Caches	Aggregation Cache	Users	Origin Server
# Instances	1	2	1	1	1
Software	HAProxy 1.5.11	ATS 5.3.0	ATS 5.3.0	Python Script	IIS 8.5
CPUs	4	6	6	6	4
RAM	4GB	6GB	6GB	6GB	4GB
NICs	2 x 10GbE (Data + Management) with 9000 MTU				
OS	Ubuntu 14.04.1 LTS x64				Win. Server 2012 R2 x64

TABLE I
VIRTUAL MACHINES (VMS)' TECHNICAL DETAILS PER INSTANCE.

G. Caching Algorithms

Caching algorithms play a crucial role on a CDN's overall performance. To leverage the content-aware demand forecasts, it is important to use algorithms that are able to benefit from that additional knowledge. Three caching algorithms, MPU [15], LRU-Weighted (LRU-W) and LFU-Weighted (LFU-W), are implemented to take advantage of the demand predictions, in addition to standard LFU, LRU, and First-In-First-Out (FIFO), which are also implemented in ATS and benchmarked. The custom LRU-W and LFU-W use the demand predictions to weight the importance of cache items. Even though other caching algorithms exist, most are either variations or combinations of the aforementioned algorithms. The algorithms' implementations are kept as similar as possible, and their behavior is cross-checked with simulations in R [15].

H. Cache Sizing

To explore the effect of different cache sizes in the experimental tests conducted, the caches are sized as fractions of the total number of unique available programs. Therefore, a cache size of 100% corresponds to a cache with the ability to hold the entire content catalog available on the 7 days testing window. Each program is assumed to require 1 storage unit.

Given its purpose of reducing the load on the Origin server and serving as an intermediate cache, the Aggregation Caches are always sized with twice the storage of the Edge Caches.

I. Key Performance Metrics

To understand the improvements provided by the envisioned content-aware OTT CDN solution, it is necessary to define key metrics by which the delivery infrastructure is evaluated.

The metrics are assessed according to how they vary along two vectors: cache size and time. By exploring variations with cache size, it is possible to evaluate the cost-benefit trade-off of increasing caches' sizes, while the variation with time is essential in Catch-up TV services with dynamic content popularity that may impact the metrics under evaluation. In order to perform the time-varying analysis, cache sizes are set at 1% of the total corpus, as defined in Section IV-H.

1) Cache Hit-Ratio: Summarizes the ratio between cache hits and cache requests, and is an indicator of how good the caching algorithm is on guessing programs that will be requested in the near-future.

2) Backend Traffic & Bandwidth: A key cost factor in content distribution is the backend traffic requirements within the CDN infrastructure before delivering the content to users. As the purpose of replica caches is to reduce the load on backend servers, a low metric is indicative of good performance.

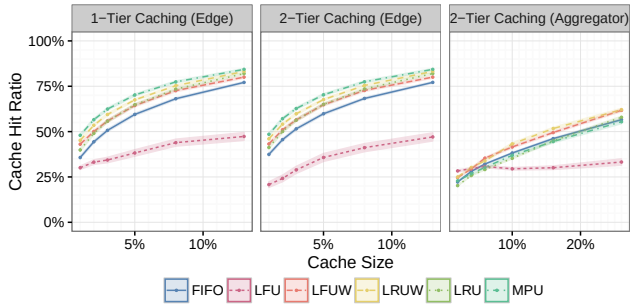
3) Request latency: The time required to service a request is an important performance indicator that must be carefully monitored to ensure that there is no impact on the user experience, as a high request latency may indicate high server or network load, which leads to queued or dropped requests.

4) QoE MOS: From a user's perspective, what matters is the service's QoE, measured through a Mean Opinion Score (MOS). Due to its subjectiveness, QoE evaluations vary significantly between users; however, objective QoE evaluation frameworks exist that provide an estimate on the expected MOS of a given service. A previously developed Smooth Streaming QoE estimation probe, presented in [31], is used to provide an objective MOS estimate.

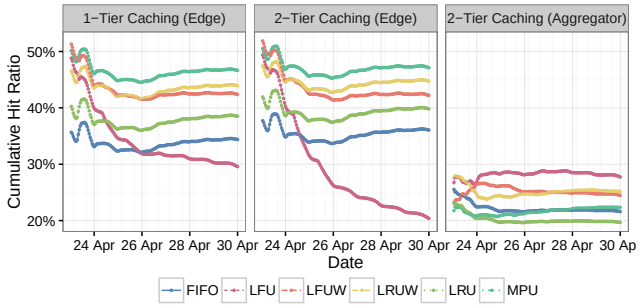
V. RESULTS AND DISCUSSION

Validating the proposed architecture is essential to draw conclusions regarding the feasibility and performance of the solution. When pertinent, the results are presented in a normalized fashion, ranging from 0% to 100%, to facilitate a graphical analysis, and the 95% Confidence Interval (CI) is shown on the average values' curve and data points.

As mentioned in Section IV-D and presented in Figure 2 two different test scenarios are evaluated that differ only on the absence – 1-Tier Caching – or presence – 2-Tier Caching – of an Aggregation Cache.

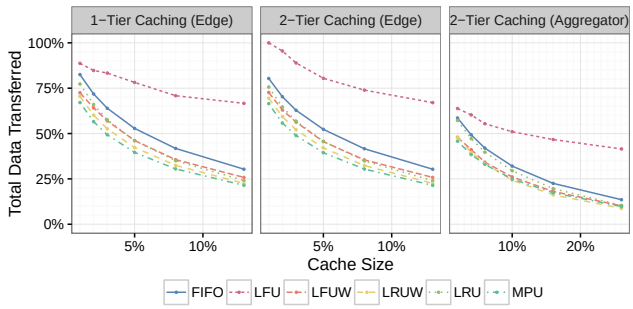


(a) Variation with Cache Size.

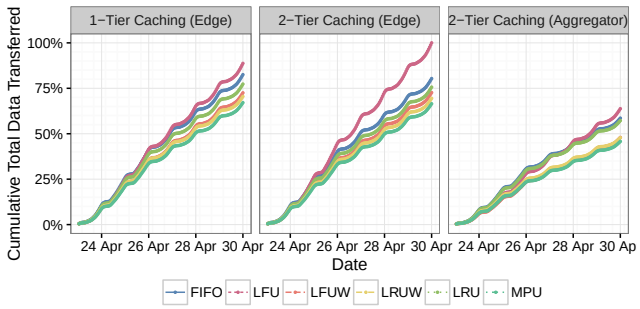


(b) Variation with Time.

Fig. 3. Cache Hit-Ratio Results.



(a) Variation with Cache Size.



(b) Variation with Time.

Fig. 4. Backend Traffic Results.

A. Cache Hit-Ratio

1) Variation with Cache Size: The first analysis explores the impact of different cache sizes in the caches' hit-ratios. The results are presented in Figure 3(a).

The results show that the usage of demand forecasts as inputs to caching algorithms, specifically to MPU, LFU-W and LRU-W, is helpful in improving the servers' caching performance, in both 1-tier and 2-tier caching scenarios – particularly for smaller cache sizes (1 to 5%). As expected, the cache hit-ratios at the edges follow similar curves, while the cache hit-ratios at the aggregation cache are significantly lower when compared to that of the edges. This behavior is due to the fact that highly popular items stay cached at the edges and are rarely requested from the aggregation cache, which instead ends up caching – and generating cache hits – for items that fall out of the edges' caches. It is interesting to observe that the best caching policy at the edges, MPU, is not necessarily also the best one to be implemented at the larger aggregation cache. Instead, LRU-W takes the lead in the aggregation cache. This behavior is caused by the interplay between the edge and aggregation caches' algorithms, which modify the traffic patterns that the aggregation tier observes. While the edges are directly serving clients, the aggregation caches' main purpose is to compensate for the edges' misses.

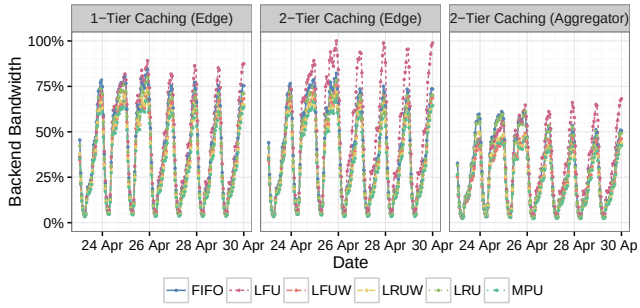
2) Evolution with Time: The results of Figure 3(b) demonstrate how caching performance varies with time. It is possible to observe that, as time progresses, some algorithms adapt better than others to content requests. As with the previous analysis, for edge caches, MPU provides the best results, closely followed by LRU-W and LFU-W, proving once more that adding content-awareness to CDNs has the potential to significantly improve their performance. In spite of excellent LFU results for the early hours of day 24, its hit-ratios' performance progressively diminishes with time, which might be explained by the effect of "cache pollution", whereby items that were initially highly popular, but lose relevance, prevent other newer items from populating the caches; hence, leading to low hit-ratios. The performance differences of LFU at the edge caches of 1-tier and 2-tier scenarios is found to be caused by the aggregation cache, which increases request latency and leads to increased pending request queuing at the edges, which in turn hinders caching performance as the items are not cached before the response is received from the server, i.e. concurrent request aggregation is not supported. The small increase in hit-ratios on all algorithms in day 24 is believed to be due to accentuated users' demand for popular content at times of the day, i.e. a result of the *superstar* effect.

As for the aggregation cache, the hit-ratios are much lower than those at the edges, with LFU taking the lead, due to its poor performance at the edges, closely followed by LFU-W and LRU-W. Overall, considering the edges and aggregation cache, MPU, LRU-W, or LFU-W yield the best performance.

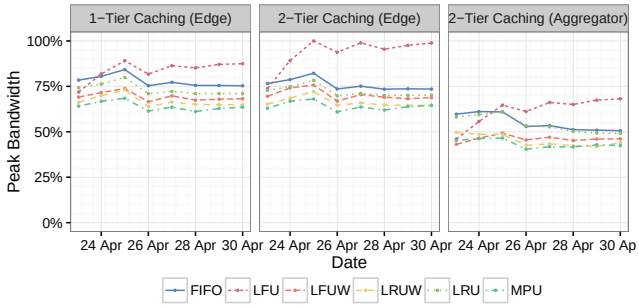
B. Backend Traffic

The volume of backend transfers is a metric that impacts the scalability and cost of CDNs. On the one hand network traffic to/from origin servers is usually expensive, while on the other hand, origin servers are often not dimensioned to be able to cope with direct demand from all users and need the fan-out capacity provided by edge and aggregation caches.

In the results, the edge caches' backend traffic is summed, thus, in 1-tier caching it represents to total amount of traffic between both edge caches and the origin server, while on the 2-

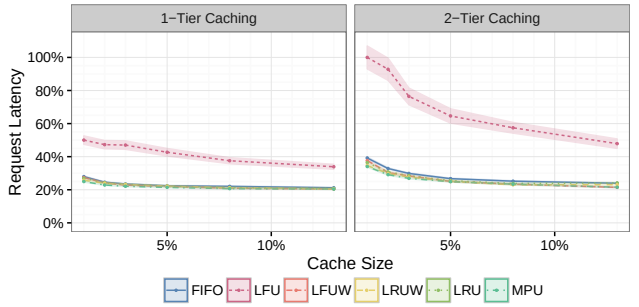


(a) Variation with Time.

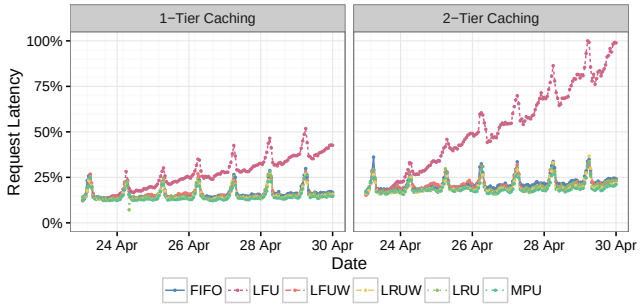


(b) Peak Variation with Time.

Fig. 5. Backend Bandwidth Results.



(a) Variation with Cache Size.



(b) Variation with Time.

Fig. 6. Request Latency.

tier scenario it shows the total traffic between the edge caches and the aggregation cache.

1) Variation with Cache Size: This metric is explored on Figure 4(a) presenting detailed information regarding the total backend traffic of each component in both 1-tier and 2-tier scenarios which, as expected, are almost identical.

It is possible to observe that the inclusion of an aggregation cache reduces the traffic to the origin server in approximately 35 % regardless of the cache algorithm chosen, in addition to aiding in edge caches' rebuilds in the event of failures.

The outcomes mirror the results presented in the previous section, and demonstrate that higher cache hit-ratios lead to a reduction in the backend data transfers, as expected.

2) Evolution with Time: As a complement to the previous study, this analysis focuses on the evolution of cumulative backend data transfers with time, which are expected to evolve inversely proportionally to the hit-ratios of each solution.

Figure 4(b) shows that, while every caching algorithm starts with approximately the same amount of data transferred, as time progresses, they quickly diverge.

The observable "wave" pattern reflects the varying content demand at the different times of day. Periods with reduced demand – late night and early mornings – show up as almost horizontal segments, while the periods with high demand are responsible for the sharp traffic increases.

At the edge caches of 1-tier and 2-tier scenarios, by the end of 7^{th} day, the best performing caching algorithm, MPU, transfers 28 to 37 % less data than the worst performing algorithm, LFU. The next best performing algorithms, LRU-W and LFU-W, only transfer $\sim$ 7 % more data than MPU, while LRU and FIFO require, respectively, $\sim$ 15 % and $\sim$ 23 % more backend data than MPU.

Analyzing the aggregation cache of the 2-tier scenario, the first conclusion is that the total data transfers performed to origin are notably lower than those of the edge caches, while the performance differences between the distinct caching algorithms are also significant, with MPU and LFU-W taking the lead, closely followed by LRU-W. The remaining traditional caching algorithms, LFU, LRU, and FIFO impose a much larger strain on the origin server's network.

C. Backend Bandwidth

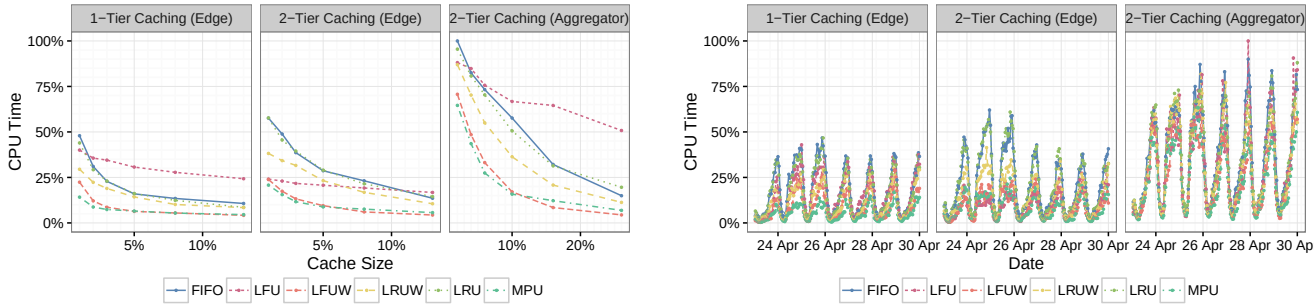
The analysis of the backend bandwidth variation with time is also pertinent, as it directly dictates how the network should be dimensioned to withstand peak demand.

1) Variation with Time: The results of Figure 5(a) illustrate how bandwidth varies throughout the day, and reveals that large discrepancies exist between minimum and maximum bandwidth requirements. The peaks match prime-time hours and are one order of magnitude larger than the minimum bandwidth requirements exhibited at late night hours.

2) Peak Variation with Time: Figure 5(b) complements the previous results by focusing on the observed peak bandwidth within each day, to demonstrate that the different caching policies directly affect bandwidth provisioning needs. From the results, it is clear that MPU, LRU-W, and LFU-W require significantly less peak bandwidth when compared to competing alternatives – LRU, FIFO, and LFU.

D. Request Latency

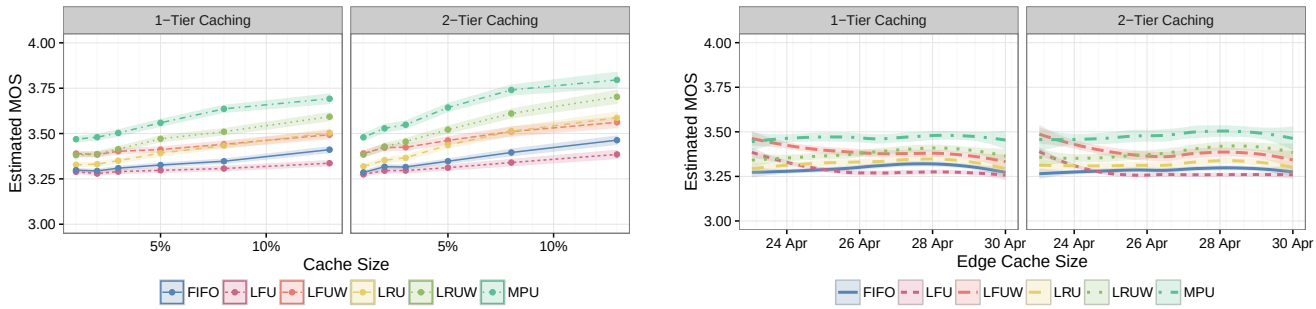
1) Variation with Cache Size: The variation of request latency with cache size is presented in Figure 6(a), where it is clearly observable that, except for LFU, all caching algorithms



(a) Variation with Cache Size.

(b) Variation with Time.

Fig. 7. CPU Time.



(a) Variation with Cache Size.

(b) Variation with Time.

Fig. 8. Estimated MOS.

present request latency metrics that fall within each other's confidence intervals. Nevertheless, a small performance lead for MPU is apparent, and it is also possible to perceive a slight overall reduction on the request latency for all caching algorithms as the caches' size increase. This effect may be attributable to better cache hit-ratios for large cache sizes.

In 2-tier scenarios, an increase on average request latency is observable due to the delay introduced by the aggregation cache. The peak average request latency exhibited by LFU in 2-tier caching scenarios corresponds to 2.3ms.

2) Evolution with Time: The results of Figure 6(b), which explore the variation of request latency with time, reveal similar conclusions to those taken on the previous analysis. LFU is clearly the worst-performing caching algorithm, with a run-away latency metric that points to severe request-queuing.

E. Total CPU Time

The total CPU time of each test is an indicator of computational demand and of energy-efficiency, as CPUs are the predominant energy consumption factor of the test platform.

1) Variation with Cache Size: The results presented in Figure 7(a) show that the total required CPU time decreases with increasing cache size – i.e. higher hit-ratios – which is explained by the lower computational demand of serving requests directly from memory instead of fetching them from the backend server and having to evict existing cache items. In both 1-tier and 2-tier scenarios MPU and LFU-W prove to be more efficient than the remaining caching algorithms

2) Evolution with Time: Figure 7(b) shows that the required CPU time is directly correlated with service demand – as seen on Figure 5(a). The load increases in high-demand periods such as prime-time. As with the results of Figure 7(a), MPU and LFU-W consistently outperform the competing caching alternatives in this metric, in 1-tier and 2-tier scenarios.

F. Impact on QoE

1) Variation with Cache Size: Figure 8(a) reveals that measurable benefits are achievable by increasing the caches' size, which is a direct impact of better cache hit-ratios, in both 1-tier and 2-tier scenarios. MPU and LRU-W provide the most significant MOS improvements over traditional caching algorithms, closely followed by LFU-W and LRU. FIFO and LFU provide the worst MOS for every considered cache size.

By adding an aggregation cache, in the 2-tier scenario, it is possible to observe that a slight MOS improvement is achievable over 1-tier scenarios for all caching algorithms.

These results demonstrate that the proposed content-aware solution is capable of boosting the performance of caching algorithms in 1-tier and 2-tier scenarios, from a technical perspective – higher hit-ratios, reduced data transfers and request latency – and from a user's perspective, in the form of a MOS enhancement.

2) Evolution with Time: The results of Figure 8(b) complement those of Figure 8(a), by showing that, with the exception of LFU-W and LFU, the performance of the remaining caching algorithms remains consistent throughout the period under analysis for both 1-tier and 2-tier scenarios. As with the previous results, MPU and LRU-W clearly dominate this evaluation and provide a significantly improved MOS when compared to the other caching algorithms.

Once again, LFU and, to some extent, LFU-W, appear to suffer with the issue of "cache pollution", which is reflected on their performance degradation with time.

VI. Conclusion

The migration of managed Catch-up TV services to OTT poses several challenges that must be addressed by next-generation delivery solutions, which must improve the services' QoE, while reducing their CAPEX and OPEX.

A content-aware architecture is proposed and thoroughly detailed that leverages machine-learning and data-mining techniques to forecast content demand, improve caching policies and facilitate autonomic resource management.

The proof-of-concept experimental implementation of the content-aware OTT delivery architecture is evaluated under realistic conditions, using request logs from a popular production Catch-up TV service to validate its design. The experimental results show that the enhanced architecture, with caching algorithms capable of taking advantage of content knowledge – MPU, LRU-W, and LFU-W –, outperform reference implementations in terms of cache hit-ratios, bandwidth savings, request latency, CPU time and users' QoE, opening the door for future, smarter, and even more efficient delivery solutions capable of leveraging content characteristics to continuously and dynamically improve themselves.

Future work will address the QoE of adaptive streaming in OTT scenarios and pre-fetching in optimized content-aware and distribution network approaches.

Acknowledgment

The authors would like to thank Fausto Carvalho (Altice Labs) and João Ferreira (MEO) for the key discussions and for providing the raw Catch-up TV dataset.

References

[1] Nielsen, "The Digital Consumer," pp. 1–28, 2014, Accessed: 09-2015. [Online]. Available: http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2014 Reports/the-digital-consumer-report-feb-2014.pdf

[2] J. Abreu *et al.*, "Survey of Catch-up TV and other time-shift services: a comprehensive analysis and taxonomy of linear and nonlinear television," *Telecommunication Systems*, mar 2016. [Online]. Available: http://dx.doi.org/10.1007/s11235-016-0157-3

[3] Cisco, "Cisco Visual Networking Index : Forecast and Methodology, 2014 - 2019," Cisco Systems, Tech. Rep., 2015, Accessed: 2015-09. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.pdf

[4] H. Abrahamsson and M. Bjorkman, "Caching for IPTV distribution with time-shift," in *2013 International Conference on Computing, Networking and Communications (ICNC)*. San Diego, CA: IEEE, Jan. 2013, pp. 916–921. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6504212

[5] G. Nencioni *et al.*, "Understanding and Decreasing the Network Footprint of Catch-up TV," in *Proceedings of the 22Nd International Conference on World Wide Web*, Rio de Janeiro, Brazil, 2013, p. 12. [Online]. Available: http://dl.acm.org/citation.cfm?id=2488388.2488472

[6] H. Koumaras *et al.*, "Media Ecosystems: A Novel Approach for Content-Awareness in Future Networks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011, vol. 6656, pp. 369–380. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-20898-0_26

[7] H. de Meer *et al.*, "Future Internet services and architectures: trends and visions," *Telecommunication Systems*, vol. 51, no. 4, pp. 219–220, dec 2012. [Online]. Available: http://dx.doi.org/10.1007/s11235-011-9430-7

[8] M. M. Amble *et al.*, "Content-aware caching and traffic management in content distribution networks," in *2011 Proceedings IEEE INFOCOM*, no. D. IEEE, apr 2011, pp. 2858–2866. [Online]. Available: http://dx.doi.org/10.1109/INFCOM.2011.5935123

[9] J. Batalla *et al.*, "Optimized decision algorithm for Information Centric Networks," *Telecommunication Systems*, vol. 61, no. 2, pp. 247–255, 2016. [Online]. Available: http://dx.doi.org/10.1007/s11235-015-9998-4

[10] M. Mangili *et al.*, "Content-aware planning models for information-centric networking," in *2014 IEEE Global Communications Conference*. IEEE, dec 2014, pp. 1854–1860. [Online]. Available: http://dx.doi.org/10.1109/GLOCOM.2014.7037078

[11] A. Tiwari and P. Kanungo, "Dynamic load balancing algorithm for scalable heterogeneous web server cluster with content awareness," in *Trendz in Information Sciences & Computing(TISC2010)*. IEEE, dec 2010, pp. 143–148. [Online]. Available: http://dx.doi.org/10.1109/TISC.2010.5714626

[12] J. Nogueira, L. Guardalben, B. Cardoso, and S. Sargento, "Catch-up TV Analytics: Statistical Characterization and Consumption Patterns Identification on a Production Service." *Multimedia Systems*, 2016. [Online]. Available: http://dx.doi.org/10.1007/s00530-016-0516-7

[13] T. Beauvisage and J.-S. Beuscart, "Audience dynamics of online catch up TV," in *Proceedings of the 21st international conference companion on World Wide Web - WWW '12 Companion*. New York, USA: ACM Press, 2012, p. 461. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2187980.2188077

[14] H. Abrahamsson and M. Nordmark, "Program popularity and viewer behaviour in a large TV-on-demand system," in *Proceedings of the 2012 ACM conference on Internet measurement conference - IMC '12*. New York, New York, USA: ACM Press, 2012, p. 199. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2398776.2398798

[15] J. Nogueira *et al.*, "Over-The-Top Catch-up TV content-aware caching," in *2016 IEEE Symposium on Computers and Communication (ISCC)*. Messina, Italy: IEEE, jun 2016, pp. 1012–1017. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7543869

[16] J. Famaey *et al.*, "Towards a predictive cache replacement strategy for multimedia content," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 219–227, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.jnca.2012.08.014

[17] L. a. Belady, "A study of replacement algorithms for a virtual-storage computer," *IBM Systems Journal*, vol. 5, no. 2, pp. 78–101, 1966. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5388441

[18] R. Ranjan *et al.*, "Cloud Resource Orchestration Programming: Overview, Issues, and Directions," *IEEE Internet Computing*, vol. 19, no. 5, pp. 46–56, sep 2015. [Online]. Available: http://dx.doi.org/10.1109/MIC.2015.20

[19] Y. Kryftis *et al.*, "Efficient entertainment services provision over a novel network architecture," *IEEE Wireless Communications*, vol. 23, no. 1, pp. 14–21, feb 2016. [Online]. Available: http://dx.doi.org/10.1109/MWC.2016.7422401

[20] R. Weingärtner *et al.*, "Cloud resource management: A survey on forecasting and profiling models," *Journal of Network and Computer Applications*, vol. 47, pp. 99–106, jan 2015. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1084804514002252

[21] J. Kephart and D. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, jan 2003. [Online]. Available: http://dx.doi.org/10.1109/MC.2003.1160055

[22] L. Watthanacheewakul, "A New Family of Transformations for Lifetime Data," in *Proceedings of the World Congress on Engineering 2014*. International Association of Engineers (IAENG), 2014, pp. 116–121. [Online]. Available: http://www.iaeng.org/publication/WCE2014/WCE2014_pp116-121.pdf

[23] M. Kuhn, "Building Predictive Models in R Using the caret Package," *Journal of Statistical Software*, vol. 28, no. 5, pp. 1–26, 2008. [Online]. Available: http://www.jstatsoft.org/v28/i05

[24] J. Szlek and A. Mendyk, "fscaret," 2015, Accessed: 09-2015. [Online]. Available: https://cran.r-project.org/web/packages/fscaret/fscaret.pdf

[25] A. Liaw, "randomForest," 2015, Accessed: 09-2015. [Online]. Available: https://cran.r-project.org/web/packages/randomForest/randomForest.pdf

[26] Microsoft Corporation, "Microsoft Smooth Streaming Protocol Specification," Microsoft Corporation, Tech. Rep., 2012, Accessed: 2015-09. [Online]. Available: http://www.iis.net/learn/media/smooth-streaming/smooth-streaming-transport-protocol

[27] NGINX Inc., "NGINX High Performance Web Server," 2015, Accessed: 12-2015. [Online]. Available: http://nginx.com

[28] D. Wessels, "Squid," 1996, Accessed: 01-2016. [Online]. Available: http://www.squid-cache.org/

[29] The Apache Software Foundation, "Apache Traffic Server," 2015, Accessed: 12-2015. [Online]. Available: http://trafficserver.apache.org

[30] HAProxy Technologies, "HAProxy - The Reliable, High Performance TCP/HTTP Load Balancer," 2015, Accessed: 12-2015. [Online]. Available: http://www.haproxy.org/

[31] A. Salvador, J. Nogueira, and S. Sargento, "QoE Assessment of HTTP Adaptive Video Streaming," 2015, pp. 235–242. [Online]. Available: http://link.springer.com/10.1007/978-3-319-18802-7_32